

The main body of the document is a large grid of 15 columns and 20 rows of small, illegible data tables or test results. Each cell in the grid contains a small table with multiple columns and rows of text, which appears to be technical data or test results. The text is too small to be read, but the layout is consistent across the entire page.

RM03

DUAL PORT LOGIC TEST PART 1
MD-11-DZRMG-A

EP-DZRMG-A-DL-A
COPYRIGHT © 1977
FICHE 2 OF 2

OCT 1977
digital
MADE IN USA

This block contains a vertical column of 16 small, illegible data tables or charts, likely representing test results for different components or conditions. Each table appears to have multiple columns and rows of data, but the text is too small to read.

11
12
13

B01

EOF1DZRMFASEQ
P82000883
DZRMG1.P11

00010000 770804
MO-11 DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1

PDP10 411

MDR1DZRMGASEQ 00010000 770804
MACY11 30(1046) 01-AUG-77 11:02 PAGE 1

01-AUG-77 10:58

.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
45
46
47
48
49
50
51
52

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRMG-A-D
PRODUCT NAME:	RMO3 DUAL PORT LOGIC TEST, PART 1
DATE CREATED:	15 JULY 77
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	DOUG RIIKONEN

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

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

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

COPYRIGHT (C) 1977, DIGITAL EQUIPMENT CORPORATION

RMO3 DUAL PORT LOGIC TEST, PART 1

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 UNIBUS & VECTOR ADDRESSES
 - 4.3 OPERATOR ACTION
- 5. OPERATING PROCEDURES
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 'SOFTWARE' SWITCH REGISTER
 - 5.3 TEST SELECTION
 - 5.4 DUAL PORT TEST CABLE CONNECTION
- 6. ERRORS
- 7. MISCELLANEOUS
 - 7.1 RESTRICTIONS
 - 7.2 LIMITATIONS
 - 7.3 EXECUTION TIME
 - 7.4 REQUIRED TESTS
 - 7.5 DISK SURFACE USAGE
 - 7.6 LOOP ON ERROR OPTION
- 8. TEST DESCRIPTIONS

53
54
55
56

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

97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152

1. ABSTRACT

THE RMO3 DUAL PORT LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RMO3 DUAL PORT LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL PORT MODE IS NOT TESTED BY THIS PROGRAM.

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 PROCESSOR
16K OF MEMORY
KW11-L OR KW11-P CLOCK
TELETYPE
RH11 OR RH70 WITH AN RMO3
RMO3 DUAL PORT TEST CABLE

2.2 PREREQUISITE PROGRAMS

RMO3 DISKLESS DIAGNOSTIC

RMO3 FUNCTIONAL TEST

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

2.3 OTHER PROGRAMS

A. THE OPERATION OF THE "PORT SELECT" SWITCH IS TESTED BY THE SECOND PART OF THE DUAL PORT LOGIC TEST.

B. DYNAMIC OPERATION OF THE DUAL PORT OPTION IS TESTED BY THE RMO3 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 BE INCLUDED IN AN 'XXDP' CHAIN.

153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
78
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
2084. STARTING PROCEDURES

4.1 STARTING ADDRESSES

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

4.2 UNIBUS & VECTOR ADDRESSES

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

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

4.3 OPERATOR ACTION

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

209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

5. OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

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

THE SWITCH SETTINGS ARE:

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

5.2 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RMO3 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

5.3 TEST SELECTION

INDIVIDUAL TESTS ARE SELECTED IN RESPONSE TO THE 'ENTER TEST NUMBER:' MESSAGE. ANY VALID TEST NUMBER CAN BE ENTERED. EACH ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN (CR). THE LOOP ON TEST SWITCH, SW<15>, 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

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320

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 RMO3 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 RMO3 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT OF THE DRIVE WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS. THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S ADDRESS PLUG.

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

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

THE TEST CABLE CONNECTION TO THE DRIVE UNDER TEST WILL DEPEND ON WHICH PROCESSOR/RH11 IS TO TEST THE DRIVE. IF THE DRIVE IS TO BE TESTED BY THE PROCESSOR ON PORT A, THE TEST CABLE IS CONNECTED FROM 'BUS A OUT' TO 'BUS B IN'. IF THE DRIVE IS TO BE TESTED BY THE PORT B PROCESSOR, THE TEST CABLE IS CONNECTED FROM 'BUS B OUT' TO 'BUS A IN'.

WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION BITS FOR PORTS A & B ARE ASSERTED IN THE SAME BIT POSITION WHEN 'RMS' (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 'RMS' (DRIVE STATUS REGISTER) TO DETERMINE THE STATE OF THE SELECTED PORTS'S ATTENTION BIT.

6. ERRORS

321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376

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

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

7. MISCELLANEOUS

7.1 RESTRICTIONS

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

7.2 LIMITATIONS

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

7.3 EXECUTION TIME

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

7.4 REQUIRED TESTS

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

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

7.5 DISK SURFACE USAGE

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

377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432

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 RMO3 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 (RMO3) 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 RMO3, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DISCREPANCY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

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

8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

THE PROGRAM VERIFIES THAT THE DRIVE HAS BEEN SEIZED BY CHECKING THE DRIVE STATUS REGISTER (RMO3) THROUGH THE SEIZING PORT AND VERIFYING THAT CORRECT STATUS IS SEEN. WHEN RMO3 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

433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488

- A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RMO3, THAT THE DRIVE IS ONLINE (RMDS1 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.

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 RMDS1.
- 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.

489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544

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 RMD51.
- B. SET VOLUME VALID AND CLEAR ANY ERROR
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.

TEST 6 PORT 'A' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 7 PORT 'B' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 10 PORT 'A' RELEASE INTERFERENCE TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
- 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 RMD51.

- 545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
- 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 RMDS1.
- B. WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RMER1 HAS NOT BEEN CLEARED.
- D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 13 PORT 'B' RELEASE W/ERRORS TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- 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.

601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656

- A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS1 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 RH70 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 RMDS1 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 RH70 AND VERIFY THAT THE DRIVE
DOES NOT RETURN TO NEUTRAL.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 16 SEIZE 'A' BY RMCS1 TEST

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

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

TEST 17 SEIZE 'B' BY RMCS1 TEST

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

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

657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712

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.

- 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.

713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768

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 RMD51.
- B. WRITE 0'S INTO RMD51 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 RMD51.
- B. WRITE 0'S INTO RMD51 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'.
- D. ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR

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

- A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
- 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.

769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824

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 RMO3.
- 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 RMO3.
- C. ISSUE A NOP COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS RESET, AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

TEST 31 RESET ATTENTION 'B' BY GO TEST

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

- A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH ATTENTION BITS ARE SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMO3.
- 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.

825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880

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 RMAS

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

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

TEST 34 PORT 'A' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'A'.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'B'.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 36 SET ATTENTION 'A' BY COMMAND TEST

881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936

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.

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

THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.

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

THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"

TEST 40 PORT 'A' SET VOLUME VALID TEST

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

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

937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992

C. RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR PORT 'B'. VERIFY THAT ATTENTION IS STILL SET AND THAT VOLUME VALID IS STILL RESET.

D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT 'B' THEN RELEASE PORT 'B'.

TEST 41 PORT 'B' SET VOLUME VALID TEST
VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.

A. WITH PORT 'B' SELECTED, RESET AND SET "UNIT READY" STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE 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 RMD51.

B. WRITE 1'S INTO RMR1 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 RMD51.

B. WRITE 1'S INTO RMR1 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

993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048

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 RMDS1.
- B. WAIT 500 MS AND READ RMDS1 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 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 RMDS1.
- B. WAIT 500 MS AND WRITE 0'B INTO RMDS1 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 RMDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMDS1 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 RMDS1.

1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104

- B. SET PORT REQUEST BY WRITING 0'S INTO RMD51 FROM PORT 'B'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

TEST 50 PORT 'A' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
- 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 RMD51.
- 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

1105
1106
1107
1108
1109
1110
1111
1112

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.

↑

1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168

;PROGRAM REVISION #001

.TITLE MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1

;*COPYRIGHT (C) 1977
 ;*DIGITAL EQUIPMENT CORP.
 ;*MAYNARD, MASS. 01754

;*PROGRAM BY D. RIIKONEN

;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
 ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

.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

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

STACK= 1100

.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL

.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS

HT= 11 ;;CODE FOR HORIZONTAL TAB
 LF= 12 ;;CODE FOR LINE FEED
 CR= 15 ;;CODE FOR CARRIAGE RETURN
 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
 PS= 177776 ;;PROCESSOR STATUS WORD

.EQUIV PS,PSW ;;STACK LIMIT REGISTER
 STKLMT= 177774 ;;PROGRAM INTERRUPT REQUEST REGISTER
 PIRQ= 177772 ;;HARDWARE SWITCH REGISTER
 DSWR= 177570 ;;HARDWARE DISPLAY REGISTER
 DDISP= 177570

;*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

001100

000011
000012
000015
000200
177776
177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

BASIC DEFINITIONS

```

1169
1170
1171      000000
1172      000040
1173      000100
1174      000140
1175      000200
1176      000240
1177      000300
1178      000340
1179

```

```

.*PRIORITY LEVEL DEFINITIONS
PRO=      0          ;; PRIORITY LEVEL 0
PR1=     40         ;; PRIORITY LEVEL 1
PR2=    100         ;; PRIORITY LEVEL 2
PR3=    140         ;; PRIORITY LEVEL 3
PR4=    200         ;; PRIORITY LEVEL 4
PR5=    240         ;; PRIORITY LEVEL 5
PR6=    300         ;; PRIORITY LEVEL 6
PR7=    340         ;; PRIORITY LEVEL 7

```

```

1180
1181      100000
1182      040000
1183      020000
1184      010000
1185      004000
1186      002000
1187      001000
1188      000400
1189      000200
1190      000100
1191      000040
1192      000020
1193      000010
1194      000004
1195      000002
1196      000001
1197

```

```

.*"SWITCH REGISTER" SWITCH DEFINITIONS
SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1

```

```

1197      .EQUIV SW09, SW9
1198      .EQUIV SW08, SW8
1199      .EQUIV SW07, SW7
1200      .EQUIV SW06, SW6
1201      .EQUIV SW05, SW5
1202      .EQUIV SW04, SW4
1203      .EQUIV SW03, SW3
1204      .EQUIV SW02, SW2
1205      .EQUIV SW01, SW1
1206      .EQUIV SW00, SW0
1207

```

```

1208
1209      100000
1210      040000
1211      020000
1212      010000
1213      004000
1214      002000
1215      001000
1216      000400
1217      000200
1218      000100
1219      000040
1220      000020
1221      000010
1222      000004
1223      000002
1224      000001

```

```

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)
BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1

```

BASIC DEFINITIONS

```

1225 .EQUIV BIT09,BIT9
1226 .EQUIV BIT08,BIT8
1227 .EQUIV BIT07,BIT7
1228 .EQUIV BIT06,BIT6
1229 .EQUIV BIT05,BIT5
1230 .EQUIV BIT04,BIT4
1231 .EQUIV BIT03,BIT3
1232 .EQUIV BIT02,BIT2
1233 .EQUIV BIT01,BIT1
1234 .EQUIV BIT00,BIT0
1235
1236 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1237 ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
1238 RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
1239 TBITVEC=14 ;: "T" BIT
1240 TRTVEC= 14 ;: TRACE TRAP
1241 BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
1242 IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
1243 PWRVEC= 24 ;: POWER FAIL
1244 EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
1245 TRAPVEC=34 ;: "TRAP" TRAP
1246 TKVEC= 60 ;: TTY KEYBOARD VECTOR
1247 TPVEC= 64 ;: TTY PRINTER VECTOR
1248 PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR
1249
1250 ;:*****
1251
1252 .SBTTL RH70 REGISTERS
1253
1254 ;:*****
1255
1256 ;CONTROL AND STATUS REGISTER 1 (RMCS1)
1257
1258 IE= 100 ;: INTERRUPT ENABLE (BIT #6)
1259 RDY= 200 ;: READY (BIT #7)
1260 A16= 400 ;: HIGH ORDER BUS ADDRESS BIT (BIT #8)
1261 A17= 100L ;: HIGH ORDER BUS ADDRESS BIT (BIT #9)
1262 PSEL= 2000 ;: PORT SELECT (BIT #10)
1263 MCPE= 20000 ;: MASSBUS PARITY ERROR (BIT #13)
1264 TRE= 40000 ;: TRANSFER ERROR (BIT #14)
1265 SC= 100000 ;: SPECIAL CONDITION (BIT #15)
1266
1267 ;WORD COUNT REGISTER (RMWC)
1268 ;(EACH BIT IS CALLED BY BIT NUMBER)
1269
1270 ;BUS ADDRESS REGISTER (RMBA)
1271 ;(EACH BIT IS CALLED BY BIT NUMBER)
1272
1273 ;CONTROL AND STATUS REGISTER 2 (RMCS2)
1274
1275 U0= 1 ;: UNIT SELECT (BIT #0)
1276 U1= 2 ;: UNIT SELECT (BIT #1)
1277 U3= 4 ;: UNIT SELECT (BIT #2)
1278 BAI= 10 ;: BUS ADDRESS INCREMENT INHIBIT (BIT #3)
1279 PAT= 20 ;: MASSBUS PARITY TEST (BIT #4)
1280 CLR= 40 ;: CLEAR (BIT #5)

```

RH70 REGISTERS

1281	000100	IR=	100	; INPUT READY (BIT #6)
1282	000200	OR=	200	; OUTPUT READY (BIT #7)
1283	000400	MDPE=	400	; MASS BUS PARITY ERROR (BIT #8)
1284	001000	MXF=	1000	; MISSED TRANSFER ERROR (BIT #9)
1285	002000	PGE=	2000	; PROGRAM ERROR (BIT #10)
1286	004000	NEM=	4000	; NON EXISTENT MEMORY (BIT #11)
1287	010000	NED=	10000	; NON EXISTENT DRIVE (BIT #12)
1288	020000	UPE=	20000	; UNIBUS PARITY ERROR (BIT #13)
1289	040000	WCE=	40000	; WRITE CHECK ERROR (BIT #14)
1290	100000	DLT=	100000	; DATA LATE (BIT #15)

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

;;*****

.SBTTL RM03 REGISTERS

;;*****

; CONTROL AND STATUS REGISTER #1. (#00)

1304	000001	GO=	1	; GO BIT (BIT #0)
1305	000002	FO=	2	; FUNCTION CODE BIT #1
1306	000004	F1=	4	; FUNCTION CODE BIT #2
1307	000010	F2=	10	; FUNCTION CODE BIT #3
1308	000020	F3=	20	; FUNCTION CODE BIT #4
1309	000040	F4=	40	; FUNCTION CODE BIT #5
1310	004000	DVA=	4000	; DEVICE AVAILABLE (BIT #11)

; CONTROL STATUS REGISTER #2 (RMCS2)

1314 000040 CLR= BITS ; CONTROLLER CLEAR

; DRIVE STATUS REGISTER (RMDS1) (#01)

1318	000001	OM	=	BIT00	; OFFSET MODE
1319	000100	VV=	100	; VOLUME VALID (BIT #6)	
1320	000200	DRY=	200	; DRIVE READY (BIT #7)	
1321	000400	DPR=	400	; DRIVE PRESENT (BIT #8)	
1322	001000	PGM=	1000	; PROGRAMABLE (BIT #9)	
1323	002000	LBT=	2000	; LAST SECTOR TRANSFERRED (BIT #10)	
1324	004000	WRL=	4000	; WRITE LOCK (BIT #11)	
1325	010000	MOL=	10000	; MEDIUM ON-LINE (BIT #12)	
1326	020000	PIP=	20000	; POSITIONING OPERATION IN PROGRESS (BIT #13)	
1327	040000	ERR=	40000	; COMPOSITE ERROR (BIT #14)	
1328	100000	ATA=	100000	; ATTENTION ACTIVE (BIT #15)	

; ERROR REGISTER #01 (RMER1) (#02)

1332	000001	ILF=	1	; ILLEGAL FUNCTION (BIT #0)
1333	000002	ILR=	2	; ILLEGAL REGISTER (BIT #1)
1334	000004	RMR=	4	; REGISTER MODIFICATION REFUSED (BIT #2)
1335	000010	PAR=	10	; PARITY ERROR (BIT #3)
1336	000020	FER=	20	; FORMAT ERROR (BIT #4)

RMO3 REGISTERS

1337	000040	WCF=	40	;WRITE CLOCK FAIL (BIT #5)
1338	000100	ECH=	100	;ECC HARD ERROR (BIT #6)
1339	000200	HCE=	200	;HEADER COMPARE ERROR (BIT #7)
1340	000400	HCRC=	400	;HEADER CRC ERROR (BIT #8)
1341	001000	AOE=	1000	;ADDRESS OVERFLOW ERROR (BIT #9)
1342	002000	IAE=	2000	;INVALID ADDRESS ERROR (BIT #10)
1343	004000	WLE=	4000	;WRITE LOCK ERROR (BIT #11)
1344	010000	DTE=	10000	;DRIVE TIMING ERROR (BIT #12)
1345	020000	OPI=	20000	;OPERATION INCOMPLETE (BIT #13)
1346	040000	UNS=	40000	;DRIVE UNSAFE (BIT #14)
1347	100000	DCK=	100000	;DATA CHECK ERROR (BIT 15)

;MAINTAINABILITY REGISTER (RMMR1) (#03)

1351	000001	DMD=	1	;DIAGINOSTIC MODE (BIT #0)
1352	001000	MUR	=	BIT09 ;MAINTENANCE UNIT READY
1353	040000	RQB	=	BIT14 ;PORT B REQUEST FLOP
1354	100000	RQA	=	BIT15 ;PORT A REQUEST FLOP

;ATTENTION SUMMARY PSEUDO-REGISTER (RMAS) (#04)

1358	000001	AT0=	1	;DEVICE 0 (BIT #0)
1359	000002	AT1=	2	;DEVICE 1 (BIT #1)
1360	000004	AT2=	4	;DEVICE 2 (BIT #2)
1361	000010	AT3=	10	;DEVICE 3 (BIT #3)
1362	000020	AT4=	20	;DEVICE 4 (BIT #4)
1363	000040	AT5=	40	;DEVICE 5 (BIT #5)
1364	000100	AT6=	100	;DEVICE 6 (BIT #6)
1365	000200	AT7=	200	;DEVICE 7 (BIT #7)

;DESIRED SECTOR/TRACK ADDRESS REGISTER (RMDA) (#05)
 ;(EACH BIT IS CALLED BY BIT NUMBER)

;DRIVE TYPE REGISTER (RMDT) (#06)

1372	000001	DT00=	1	;DRIVE TYPE NUMBER BIT 1
1373	000002	DT01=	2	;DRIVE TYPE NUMBER BIT 2
1374	000004	DT02=	4	;DRIVE TYPE NUMBER BIT 3
1375	000010	DT03=	10	;DRIVE TYPE NUMBER BIT 4
1376	000020	DT04=	20	;DRIVE TYPE NUMBER BIT 5
1377	000040	DT05=	40	;DRIVE TYPE NUMBER BIT 6
1378	000100	DT06=	100	;DRIVE TYPE NUMBER BIT 7
1379	000200	DT07=	200	;DRIVE TYPE NUMBER BIT 8
1380	000400	DT08=	400	;DRIVE TYPE NUMBER BIT 9
1381	004000	DRQ=	4000	;DRIVE REQUEST REQUIRED (BIT #11)
1382	020000	MOH=	20000	;MOVING HEAD (BIT #13)
1383	040000	TAP=	40000	;TAPE DRIVE (BIT #14)
1384	100000	NBA=	100000	;NOT BLOCK ADDRESSED (BIT #15)

;LOOK-AHEAD REGISTER (RMLA) (#07)

1388	000100	SC0=	100	;SECTOR COUNT FIELD 0 (BIT #6)
1389	000200	SC1=	200	;SECTOR COUNT FIELD 1 (BIT #7)
1390	000400	SC2=	400	;SECTOR COUNT FIELD 2 (BIT #8)
1391	001000	SC3=	1000	;SECTOR COUNT FIELD 3 (BIT #9)
1392	002000	SC4=	2000	;SECTOR COUNT FIELD 4 (BIT #10)

RMD3 REGISTERS

```

1393
1394
1395
1396           000010           DPE=      10           ;DATA PARITY ERR R (BIT #3)
1397           000200           DVC=     200           ;DEVICE CHECK (BIT #7)
1398           002000           LBC=    2000           ;LOSS OF BIT CLOCK (BIT #10)
1399           004000           LSC=    4000           ;LOSS OF SYSTEM CLOCK (BIT #11)
1400           010000           IVC=   10000           ;INVALID COMMAND (BIT #12)
1401           020000           OPE=   20000           ;OPERATOR ERROR (BIT #13)
1402           100000           SKI=   10000           ;SEEK INCOMPLETE (BIT #14)
1403
1404           ;OFFSET REGISTER (RMOF) (#11)
1405
1406           000200           OFD=     200           ;OFFSET FORWARD (BIT #5)
1407           002000           HCI=    2000           ;HEADER COMPARE INHIBIT (BIT #10)
1408           004000           ECI=    4000           ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1409           010000           FMT16= 10000           ;FORMAT BIT (BIT #12)
1410
1411           ;DESIRED CYLINDER ADDRESS (RMDC) (#12)
1412           ;(EACH BIT IS CALLED BY BIT NUMBER)
1413
1414           ;SERIAL NUMBER REGISTER (RMSN) (#14)
1415           ;(EACH IS CALLED BY BIT NUMBER)
1416
1417           ;ECC POSITION REGISTER (RMEC1) (#16)
1418           ;(EACH BIT IS CALLED BY BIT NUMBER)
1419
1420           ;ECC PATTERN REGISTER (RMEC2) (#17)
1421           ;(EACH BIT IS CALLED BY BIT NUMBER)
1422
1423           ;;*****
1424           .SBTTL DEFINITIONS OF THE RH70/RMD3 ADDRESS INDEXES
1425
1426           ;;*****
1427
1428           000000           RMCS1=0           ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
1429           000002           RMWC=2           ;WORD COUNT REGISTER (NOT A DRIVE REG)
1430           000004           RMBA=4           ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
1431           000006           RMDA=6           ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
1432           000010           RMCS2=10          ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
1433           000012           RMDS1=12          ;DRIVE STATUS REGISTER (DRIVE REG 01)
1434           000014           RMER1=14          ;ERROR REGISTER #1 (DRIVE REG. 02)
1435           000016           RMAS=16           ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
1436           000020           RMLA=20           ;LOOK AHEAD REGISTER (DRIVE REG. 07)
1437           000022           RMDB=22           ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
1438           000024           RMMR1=24          ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
1439           000026           RMDT=26           ;DRIVE TYPE REGISTER (DRIVE REG. 06)
1440           000030           RMSN=30           ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
1441           000032           RMOF=32           ;OFFSET REGISTER (DRIVE REG. 11)
1442           000034           RMDC=34           ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
1443           000040           RMMR2=40          ;MAINTENANCE REGISTER #2 (DRIVE REG. 14)
1444           000042           RMER2=42          ;ERROR REGISTER #2 (DRIVE REG. 15)
1445           000044           RMEC1=44          ;ECC POSITION REGISTER (DRIVE REG. 16)
1446           000046           RMEC2=46          ;ECC PATTERN REGISTER (DRIVE REG. 17)
1447
1448

```

.SBTTL TRAP CATCHER

1449
1450
1451 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

1455 000174 000174
1456 000174 000000
1457 000176 000000

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

.SBTTL ACT11 HOOKS

1458
1459
1460
1461
1462

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

1463 000200
1464 000046
1465 000046 065726
1466 000052
1467 000052 020000
1468 000200

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

.SBTTL STARTING ADDRESS = 200

1470
1471
1472 000200 000137 002160
1473

JMP START ; ; START THE PROGRAM

.SBTTL START THE PROGRAM AND CHANGE THE RH70 ADDRESS = 204

1474
1475
1476 000204 000137 002166
1477
1478

JMP START1 ; ; START AND CHANGE THE RH70 ADDRESS

COMMON TAGS

.SBTTL COMMON TAGS

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

Address	Hex	Hex	Label	Type	Value	Description
1479			.SBTTL	COMMON TAGS		
1485	001100		SCMTAG:		.=1100	START OF COMMON TAGS
1486	001100	000000	\$PASS:	.WORD	0	CONTAINS PASS COUNT
1487	001100	000000	\$STNM:	.BYTE	0	CONTAINS THE TEST NUMBER
1488	001102	000	\$ERFLG:	.BYTE	0	CONTAINS ERROR FLAG
1489	001103	000	\$ICNT:	.WORD	0	CONTAINS SUBTEST ITERATION COUNT
1490	001104	000000	\$LPADR:	.WORD	0	CONTAINS SCOPE LOOP ADDRESS
1491	001106	000000	\$LPERR:	.WORD	0	CONTAINS SCOPE RETURN FOR ERRORS
1492	001110	000000	\$ERTTL:	.WORD	0	CONTAINS TOTAL ERRORS DETECTED
1493	001112	000000	\$ITEMB:	.BYTE	0	CONTAINS ITEM CONTROL BYTE
1494	001114	000	\$ERMAX:	.BYTE	1	CONTAINS MAX. ERRORS PER TEST
1495	001115	001	\$ERRPC:	.WORD	0	CONTAINS PC OF LAST ERROR INSTRUCTION
1496	001116	000000	\$GDADR:	.WORD	0	CONTAINS ADDRESS OF 'GOOD' DATA
1497	001120	000000	\$BDADR:	.WORD	0	CONTAINS ADDRESS OF 'BAD' DATA
1498	001122	000000	\$GDAT:	.WORD	0	CONTAINS 'GOOD' DATA
1499	001124	000000	\$BDAT:	.WORD	0	CONTAINS 'BAD' DATA
1500	001126	000000				RESERVED--NOT TO BE USED
1501	001130	000000				
1502	001132	000000				
1503	001134	000	\$AUTOB:	.BYTE	0	AUTOMATIC MODE INDICATOR
1504	001135	000	\$INTAG:	.BYTE	0	INTERRUPT MODE INDICATOR
1505	001136	000000				
1506	001140	177570	\$SWR:	.WORD	DSWR	ADDRESS OF SWITCH REGISTER
1507	001142	177570	\$DISPLAY:	.WORD	DDISP	ADDRESS OF DISPLAY REGISTER
1508	001144	177560	\$TKS:		177560	TTY KBD STATUS
1509	001146	177562	\$TKB:		177562	TTY KBD BUFFER
1510	001150	177564	\$TPS:		177564	TTY PRINTER STATUS REG. ADDRESS
1511	001152	177566	\$TPB:		177566	TTY PRINTER BUFFER REG. ADDRESS
1512	001154	000	\$NULL:	.BYTE	0	CONTAINS NULL CHARACTER FOR FILLS
1513	001155	002	\$FILLS:	.BYTE	2	CONTAINS # OF FILLER CHARACTERS REQUIRED
1514	001156	012	\$FILLC:	.BYTE	12	INSERT FILL CHARS. AFTER A "LINE FEED"
1515	001157	000	\$TPFLG:	.BYTE	0	"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1516	001160	000000	\$REGAD:	.WORD	0	CONTAINS THE ADDRESS FROM WHICH (\$REGAD) WAS OBTAINED
1517						
1518	001162	000000	\$REGO:	.WORD	0	CONTAINS ((\$REGAD)+0)
1519	001164	000000	\$TMP0:	.WORD	0	USER DEFINED
1520	001166	000000	\$TMP1:	.WORD	0	USER DEFINED
1521	001170	000000	\$TMP2:	.WORD	0	USER DEFINED
1522	001172	000000	\$TMP3:	.WORD	0	USER DEFINED
1523	001174	000000	\$TMP4:	.WORD	0	USER DEFINED
1524	001176	000000	\$TIMES:		0	MAX. NUMBER OF ITERATIONS
1525	001200	000000	\$ESCAPE:		0	ESCAPE ON ERROR ADDRESS
1526	001202	177607	\$BELL:	.ASCIZ	<207><377><377>	CODE FOR BELL
1527	001206	077	\$QUES:	.ASCII	/?/	QUESTION MARK
1528	001207	015	\$CRLF:	.ASCII	<15>	CARRIAGE RETURN
1529	001210	000012	\$LF:	.ASCIZ	<12>	LINE FEED
1530						
1531		000015	CR	=	15	
1532		000012	LF	=	12	
1533	001212	172540	\$LKCSR:	.WORD	172540	ADDR OF KW11-P STATUS REGISTER
1534	001214	172542	\$LKCSB:	.WORD	172542	ADDR OF KW11-P COUNTER BUFFER

000377

1535	001216	000104	SLPVEC: .WORD	104	; ADDR OF KW11-P VECTOR
1536	001220	177546	SLKS: .WORD	177546	; ADDR OF KW11-L STATUS REGISTER
1537	001222	000100	SLIVEC: .WORD	100	; ADDR OF KW11-L VECTOR
1538	001224	000000	PORTA: .WORD	0	; ADDRESS OF PORT A
1539	001226	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
1540	001230	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
1541	001232	000000	RQSTA: .WORD	0	; REQUEST BIT FOR PORT A
1542	001234	000000	RQSTB: .WORD	0	; REQUEST BIT FOR PORT B
1543	001236	000000	ASR1: .WORD	0	; ATA-A OR ATA-B = 1
1544	001240	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
1545	001242	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
1546	001244	000000	OPPR: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
1547	001246	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
1548	001250	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURRED
1549	001252	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
1550	001254	000000	RELERR: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
1551	001256	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
1552	001260	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
1553	001262	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
1554	001264	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
1555	001266	000000	TIMEAM: .WORD	0	; PORT A TIMEOUT VALUE - 25%
1556	001270	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
1557	001272	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
1558	001274	000000	TIMEBM: .WORD	0	; PORT B TIME VALUE - 25%
1559	001276	000000	TIMES: .WORD	0	; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
1560	001300	000000	KYBCTL: .WORD	0	; SINGLE TEST INDICATOR
1561	001302	000000	CHGADR: .WORD	0	; CHANGE THE RH70 ADDRESS INDICATOR

1562
1563 ;*****

1564
1565 .SBTTL RH70/RM03 UNIBUS AND VECTOR ADDRESSES

1566
1567 ;*****

1568					
1569	001304	176700	\$RMADR: .WORD	176700	; RH70/RM03 UNIBUS ADDRESS
1570	001306	000254	\$RMVEC: .WORD	254	; RH70 INTERRUPT VECTOR ADDRESS

1571

ERROR POINTER TABLE

.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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

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

\$ERRTB:

;ERROR 1

;WRONG DRIVE TYPE

1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586 001310
1587
1588
1589
1590 001310 072014
1591 001312 076435
1592 001314 100326
1593 001316 100614
1594
1595
1596
1597 001320 072035
1598 001322 076435
1599 001324 100326
1600 001326 100614
1601
1602
1603
1604 001330 072057
1605 001332 076506
1606 001334 100342
1607 001336 100614
1608
1609
1610
1611 001340 072141
1612 001342 076555
1613 001344 100410
1614 001346 100627
1615
1616
1617
1618 001350 072172
1619 001352 076700
1620 001354 100356
1621 001356 100621
1622
1623
1624
1625 001360 072240
1626 001362 077150
1627 001364 100430

EM1
DH1
DT1
DF1

;ERROR 2

;DRIVE NOT ON LINE

EM2
DH1
DT1
DF1

;ERROR 3

;SERIAL NUMBERS NOT THE SAME

EM3
DH3
DT3
DF1

;ERROR 4

;DRIVE NOT SEIZED BY PORT 'N'

EM4
DH4
DT7
DF7

;ERROR 5

;WRONG STATUS SEEN BY THE SEIZING PORT

EM5
DH5
DT5
DF5

;ERROR 6

;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA

EM6
DH13
DT13

ERROR POINTER TABLE

1628	001366	100621	DF5	
1629				
1630				;ERROR 7
1631				
1632	001370	072340	EM7	;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1633	001372	076754	DH7	
1634	001374	100410	DT7	
1635	001376	100627	DF7	
1636				
1637				;ERROR 10
1638				
1639	001400	072421	EM10	;REGISTER CONTENTS INCORRECT
1640	001402	076700	DH5	
1641	001404	100356	DT5	
1642	001406	100621	DF5	
1643				
1644				;ERROR 11
1645				
1646	001410	072451	EM11	;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1647	001412	077077	DH11	
1648	001414	100326	DT1	
1649	001416	100614	DF1	
1650				
1651				;ERROR 12
1652				
1653	001420	072535	EM12	;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1654	001422	077644	DH36	
1655	001424	100516	DT37	
1656	001426	100642	DF36	
1657				
1658				;ERROR 13
1659				
1660	001430	072605	EM13	; 'VOLUME VALID' BIT NOT SET BY READIN PRESET
1661	001432	077150	DH13	
1662	001434	100430	DT13	
1663	001436	100621	DF5	
1664				
1665				;ERROR 14
1666				
1667	001440	072672	EM14	; 'VOLUME VALID' SET ON THE OPPOSITE PORT
1668	001442	077150	DH13	
1669	001444	100430	DT13	
1670	001446	100621	DF5	
1671				
1672				;ERROR 15
1673				
1674	001450	072735	EM15	; THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1675	001452	076754	DH7	
1676	001454	100410	DT7	
1677	001456	100627	DF7	
1678				
1679				;ERROR 16
1680				
1681	001460	073014	EM16	;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1682	001462	076754	DH7	
1683	001464	100410	DT7	

1684	001466	100627	DF7	
1685				
1686				;ERROR 17
1687				
1688	001470	073067	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
1689	001472	076754	DH7	
1690	001474	100410	DT7	
1691	001476	100627	DF7	
1692				
1693				;ERROR 20
1694				
1695	001500	073146	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
1696	001502	077644	DH36	
1697	001504	100516	DT37	
1698	001506	100642	DF36	
1699				
1700				;ERROR 21
1701				
1702	001510	073226	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
1703	001512	077644	DH36	
1704	001514	100516	DT37	
1705	001516	100642	DF36	
1706				
1707				;ERROR 22
1708				
1709	001520	073301	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
1710	001522	077270	DH22	
1711	001524	100446	DT22	
1712	001526	100636	DF31	

1713				
1714				
1715				
1716	001530	073366	EM23	
1717	001532	077366	DH23	
1718	001534	100460	DT23	

;ERROR 23

;TIMEOUT CLEARED THE DRIVE'S ERROR BIT

ERROR POINTER TABLE

1719	001536	100614	DF1	
1720				
1721			;ERROR 24	
1722				
1723	001540	073434	EM24	;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1724	001542	077366	DH23	
1725	001544	100460	DT23	
1726	001546	100614	DF1	
1727				
1728				
1729			;ERROR 25	
1730				
1731	001550	073513	EM25	;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1732	001552	077644	DH36	
1733	001554	100506	DT36	
1734	001556	100642	DF36	
1735				
1736				
1737			;ERROR 26	
1738				
1739	001560	073556	EM26	;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET

K03

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

ERROR POINTER TABLE

MACY11 30(1046)

01-AUG-77 11:02 PAGE 36

SEQ 0038

1740	001562	077270	DH22	
1741	001564	100446	DT22	
1742	001566	100636	DF31	
1743				
1744				;ERROR 27
1745				
1746	001570	073643	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1747	001572	076754	DH7	
1748	001574	100410	DT7	
1749	001576	100627	DF7	
1750				
1751				;ERROR 30
1752				
1753	001600	073721	EM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1754	001602	077644	DH36	
1755	001604	100506	DT36	
1756	001606	100642	DF36	
1757				
1758				;ERROR 31
1759				
1760	001610	074016	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SE
1761	001612	077545	DH31	
1762	001614	100474	DT31	
1763	001616	100636	DF31	
1764				
1765				;ERROR 32
1766				
1767	001620	074073	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1768	001622	076700	DH5	
1769	001624	100356	DT5	
1770	001626	100621	DF5	
1771				
1772				;ERROR 33
1773				
1774	001630	074144	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1775	001632	077644	DH36	
1776	001634	100506	DT36	
1777	001636	100642	DF36	
1778				
1779				;ERROR 34
1780				
1781	001640	074246	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1782	001642	077644	DH36	
1783	001644	100506	DT36	
1784	001646	100642	DF36	
1785				
1786				;ERROR 35
1787				
1788	001650	074351	EM35	;DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT 0
1789	001652	077644	DH36	

1790	001654	100516	DT37	
1791	001656	100642	DF36	
1792				
1793				;ERROR 36
1794				
1795	001660	074430	EM36	;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
1796	001662	077644	DH36	
1797	001664	100506	DT36	
1798	001666	100642	DF36	
1799				
1800				;ERROR 37
1801				
1802	001670	074502	EM37	;DRIVE IS NON-EXISTENT
1803	001672	077644	DH36	
1804	001674	100516	DT37	
1805	001676	100642	DF36	
1806				
1807				;ERROR 40
1808				
1809	001700	074550	EM40	;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1810	001702	076435	DH1	
1811	001704	100460	DT23	
1812	001706	100614	DF1	
1813				
1814				;ERROR 41
1815				
1816	001710	074625	EM41	;TIMEOUT CLEARED ATTENTION BIT
1817	001712	077366	DH23	
1818	001714	100460	DT23	
1819	001716	100614	DF1	
1820				
1821				;ERROR 42
1822				
1823	001720	074667	EM42	;DRIVE NOT IN NEUTRAL OR SEIZED
1824	001722	077673	DH42	
1825	001724	100526	DT42	
1826	001726	100645	DF42	
1827				
1828				;ERROR 43
1829				
1830	001730	074755	EM43	;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1831	001732	077673	DH42	
1832	001734	100526	DT42	
1833	001736	100645	DF42	
1834				
1835				;ERROR 44
1836				
1837	001740	075032	EM44	;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1838	001742	077712	DH44	
1839	001744	100474	DT31	
1840	001746	100636	DF31	
1841				
1842				;ERROR 45
1843				
1844	001750	075107	EM45	;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1845	001752	076435	DH1	

ERROR POINTER TABLE

1846	001754	100326	BT1	
1847	001756	100614	DF1	
1848				
1849				;ERROR 46
1850				
1851	001760	075166	EM46	;CAN'T ACCESS DRIVE THROUGH EITHER PORT
1852	001762	100010	DH46	
1853	001764	100534	DT46	
1854	001766	100636	DF31	
1855				
1856				;ERROR 47
1857				
1858	001770	075235	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
1859	001772	077366	DH23	
1860	001774	100460	DT23	
1861	001776	100614	DF1	
1862				
1863				;ERROR 50
1864				
1865	002000	075323	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMA
1866	002002	077150	DH13	
1867	002004	100430	DT13	
1868	002006	100621	DF5	
1869				
1870				;ERROR 51
1871				
1872	002010	075405	EM51	;ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1873	002012	076700	DH5	
1874	002014	100356	DT5	
1875	002016	100621	DF5	
1876				
1877				;ERROR 52
1878				
1879	002020	075474	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1880	002022	077150	DH13	
1881	002024	100430	DT13	
1882	002026	100621	DF5	
1883				
1884				;ERROR 53
1885				
1886	002030	075567	EM53	
1887	002032	077356	DH23	
1888	002034	100326	DT1	
1889	002036	100614	DF1	
1890				
1891				;ERROR 54
1892				
1893	002040	075650	EM54	;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1894	002042	077270	DH22	
1895	002044	100546	DT54	
1896	002046	100636	DF31	
1897				
1898				;ERROR 55
1899				
1900	002050	075743	EM55	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1901	002052	100106	DH55	

ERROR POINTER TABLE

1902	002054	100560	DT55	
1903	002056	100647	DF55	
1904				
1905				;ERROR 56
1906				
1907	002060	076010	EM56	;RH11 DIDN'T RESPOND TO ADDRESSING
1908	002062	100164	DH56	
1909	002064	100572	DT56	
1910	002066	100653	DF56	
1911				
1912				
1913				;ERROR 57
1914	002070	076052	EM57	;PORT REQUEST FLOPS WRONG
1915	002072	100173	DH57	
1916	002074	100576	DT57	
1917	002076	100621	DF57	
1918				
1919				;ERROR 60
1920	002100	076113	EM60	;ATTENTION BITS NOT RESET BY RMAS
1921	002102	076700	DH5	
1922	002104	100356	DT5	
1923	002106	100621	DF5	
1924				
1925				;ERROR 61
1926	002110	076157	EM61	;ATTENTION NOT RESET BY GO
1927	002112	077366	DH23	
1928	002114	100460	DT23	
1929	002116	100614	DF1	
1930				
1931				;ERROR 62
1932	002120	076211	EM62	;ATTENTION RESET BY GO WHEN NOT SEIZED
1933	002122	077150	DH13	
1934	002124	100430	DT13	
1935	002126	100621	DF5	
1936				
1937				;ERROR 63
1938	002130	076257	EM63	;DRIVE SEIZED BY UNIT READY CHANGE
1939	002132	077644	DH36	
1940	002134	100506	DT36	
1941	002136	100642	DF36	
1942				
1943				;ERROR 64
1944	002140	076321	EM64	;ATTENTION NOT SET BY UNIT READY CHANGE
1945	002142	076754	DH7	
1946	002144	100410	DT7	
1947	002146	100627	DF7	
1948				
1949				;ERROR 65
1950	002150	076370	EM65	;VV NOT RESET BY UNIT READY
1951	002152	076700	DH5	
1952	002154	100356	DT5	
1953	002156	100621	DF5	
1954				
1955				
1956				;*****
1957				

```

1958 .SBTTL STARTUP AND INITIALIZATION ROUTINES
1959
1960 ;;*****
1961
1962 START: CLR CHGADR ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
1963 BR START2 ;GO TO THE START
1964 START1: MOV #-1,CHGADR ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
1965 START2: RESET ;CLEAR THE BUS
1966 .SBTTL INITIALIZE THE COMMON TAGS
1967 ;;CLEAR THE COMMON TAGS (SCHTAG) AREA
1968 MOV #SCHTAG,R6 ;FIRST LOCATION TO BE CLEARED
1969 CLR (R6)+ ;CLEAR MEMORY LOCATION
1970 CMP #SWR,R6 ;;DONE?
1971 BNE -6 ;;LOOP BACK IF NO
1972 MOV #STACK,SP ;SETUP THE STACK POINTER
1973 ;;INITIALIZE A FEW VECTORS
1974 MOV #SCOPE,@IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
1975 MOV #340,@IOTVEC+2 ;LEVEL 7
1976 MOV #ERROR,@EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
1977 MOV #340,@EMTVEC+2 ;LEVEL 7
1978 MOV #TRAP,@TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
1979 MOV #340,@TRAPVEC+2 ;LEVEL 7
1980 MOV SENDCT,SEOPCT ;SETUP END-OF-PROGRAM COUNTER
1981 CLR $TIMES ;INITIALIZE NUMBER OF ITERATIONS
1982 CLR $ESCAPE ;CLEAR THE ESCAPE ON ERROR ADDRESS
1983 MOV #1,$ERMAX ;ALLOW ONE ERROR PER TEST
1984 MOV #,$SLPADR ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1985 MOV #,$SLPERR ;SETUP THE ERROR LOOP ADDRESS
1986 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1987 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1988 MOV @ERRVEC,-(SP) ;SAVE ERROR VECTOR
1989 MOV #64$,@ERRVEC ;SET UP ERROR VECTOR
1990 MOV #DSWR,$SWR ;SETUP FOR A HARDWARE SWICH REGISTER
1991 MOV #DDISP,$DISPLAY ;AND A HARDWARE DISPLAY REGISTER
1992 CMP #-1,$SWR ;TRY TO REFERENCE HARDWARE SWR
1993 BNE 66$ ;BRANCH IF NO TIMEOUT TRAP OCCURRED
1994 ;AND THE HARDWARE SWR IS NOT = -1
1995 BR 65$ ;BRANCH IF NO TIMEOUT
1996 MOV #65$,(SP) ;SET UP FOR TRAP RETURN
1997 RTI
1998 MOV #SWREG,$SWR ;POINT TO SOFTWARE SWR
1999 MOV #DISPREG,$DISPLAY
2000 MOV (SP)+,@ERRVEC ;RESTORE ERROR VECTOR
2001
2002 INC #-1 ;FIRST START ?
2003 BNE 1$ ;BR IF NOT
2004 TYPE TITLE ;TYPE PROGRAM NAME
2005 JSR PC,$TKINT ;SETUP THE TTY KEYBOARD
2006 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
2007 TST @42 ;ARE WE RUNNING UNDER XXDP/ACT?
2008 BNE 67$ ;BRANCH IF YES
2009 CMP $SWR,$SWREG ;SOFTWARE SWITCH REG SELECTED?
2010 BNE 68$ ;BRANCH IF NO
2011 GTSWR ;GET SOFT-SWR SETTINGS
2012 BR 68$
2013 MOV #1,$AUTOB ;SET AUTO-MODE INDICATOR

```

```

2014 002456          68$: JSR      PC,CHANGE      ;CHECK/CHANGE THE RH11 ADDRESS
2015 002456 004737 003050 TYPE      ,ENTERA      ;ENTER DRIVE ADDRESS
2016 002462 104401 071425 RDOCT          ;GET THE ADDRESS
2017 002466 104412          MOV      (SP)+,PORTA    ;STORE THE ADDRESS
2018 002470 012637 001224          CMP      PORTA,#7      ;SEE IF ADDRESS TOO LARGE
2019 002474 023727 001224 000007 BLOS     2$           ;BR IF NOT
2020 002502 101403          TYPE      ,ADRERR      ;TYPE ADDRESS ERROR MESSAGE
2021 002504 104401 071455          BR       1$           ;TRY AGAIN
2022 002510 000744          MOV      PORTA,PORTB   ;GENERATE THE PORT B ADDRESS
2023 002512 013737 001224 001226 2$: INC      PORTB        ;INCREMENT THE ADDRESS
2024 002520 005237 001226          BIC     #16,PORTB     ;LEAVE BIT 0
2025 002524 042737 000016 001226 MOV      PORTA,-(SP)   ;PUT PORT A ADDRESS ON THE STACK
2026 002532 013746 001224          BIC     #16,(SP)      ;SAVE BITS 1 & 2
2027 002536 042716 177771          BIS     (SP)+,PORTB   ;SET BITS 1 & 2 IN PORT B ADDRESS
2028 002542 052637 001226          TYPE      ,PORTAIS   ;'PORT A ADDRESS IS '
2029 002546 104401 071477          MOV      PORTA,-(SP) ;SAVE PORTA FOR TYPEOUT
2030 002552 013746 001224          TYPE     PORTA        ;TYPE PORT A ADDRESS
2031          GO     TYPE--OCTAL ASCII
2032 002556 104403          TYPE     1            ;TYPE 1 DIGIT(S)
2033 002560          .BYTE 1
2034 002561          .BYTE 0
2035 002562 104401 071525          TYPE     ,PORTBIS   ;'PORT B ADDRESS IS '
2036 002566 013746 001226          MOV      PORTB,-(SP) ;SAVE PORTB FOR TYPEOUT
2037          TYPE     PORTB        ;TYPE PORT B ADDRESS
2038          GO     TYPE--OCTAL ASCII
2039 002572 104403          TYPE     1            ;TYPE 1 DIGIT(S)
2040 002574          .BYTE 1
2041 002575          .BYTE 0
2042 002576 104401 001207          TYPE     ,$CRLF     ;ANOTHER CR-LF
2043 002602 013737 001224 001230 MOV      PORTA,PORTC  ;GENERATE ADDRESS OF DRIVE NOT TESTED
2044 002610 062737 000006 001230 ADD      #6,PORTC     ;COMPLEMENT SOME BITS
2045 002616 042737 177770 001230 BIC     #17,PORTC     ;SAVE ONLY LOWER BITS
2046 002624 013701 001224          MOV      PORTA,R1    ;USE PORT A ADDRESS AS INDEX
2047 002630 116137 100770 001236 MOVVB   ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
2048 002636 005037 001262          CLR     TIMEA        ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2049 002642 005037 001264          CLR     TIMEAP       ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2050 002646 005037 001270          CLR     TIMEB        ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2051 002652 005037 001272          CLR     TIMEBP       ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2052 002656 004737 065746          JSR     PC,CKCLK     ;SETUP CLOCK
2053 002662 000137 002676          JMP     EXEC         ;CLOCK HAS BEEN STARTED
2054 002666 104401 071553          TYPE     ,NOCLOCK   ;NO CLOCK ON SYSTEM
2055 002672 000000 3$: HALT          ;FATAL ERROR
2056 002674 000776          BR      3$          ;INTERLOCK THE HALT
2057          ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
2058          EXEC:
2059 002676 000005          RESET          ;CLEAR EVERYTHING
2060 002700 005037 177776          CLR     PS           ;CLEAR THE PROCESSOR STATUS WORD
2061 002704 104401 001207          TYPE     , $CRLF    ;CR-LF
2062 002710 013700 001304          MOV     $RADDR,R0   ;RH11 ADDRESS FOR INDEXING
2063 002714 012706 001100          MOV     #STACK,SP   ;LOAD STACK POINTER
2064 002720 004737 065746          JSR     PC,CKCLK     ;START THE CLOCK
2065 002724 000240          NOP            ;RETURN IF NO CLOCK
2066 002726 004737 067644          JSR     PC,$TKINT    ;INITIALIZE THE KEYBOARD
2067 002732 005037 001300          CLR     KYBCTL      ;CLEAR SINGLE TEST INDICATOR
2068 002736 005037 001100          CLR     $PASS       ;CLEAR THE PASS COUNT
2069 002742 112737 000001 001115 MOVVB   #1,$ERMAX    ;SET ERROR MAX TO 1

```

```

2070 002750 012737 002750 001106      MOV      #.,SLPADR      ;INITIAL SETTING FOR LOOP ADDRESS
2071 002756 012737 002756 001110      MOV      #.,SLPERR     ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
2072 002754 104401 071622      1$:     TYPE      ,TESTNO ;ASK FOR TEST NUMBER
2073 002770 104412      RDOCT      ;GET THE NUMBER
2074 002772 012601      MOV      (SP)+,R1     ;PUT ENTRY INTO R1
2075 002774 001002      BNE      2$          ;BR IF NOT ZERO
2076 002776 000137 003164      JMP      TST1        ;ENTER ZERO - PERFORM ALL TESTS
2077 003002 020137 101000      2$:     CMP      R1,MAXTN ;SEE IF NUMBER GREATER THAN MAXIMUM
2078 003006 003403      BLE      3$          ;BR IF LESS OR EQUAL
2079 003010 104401 071642      TYPE      ,BADNO     ;BAD ENTRY
2080 003014 000763      BR       1$          ;TRY AGAIN
2081 003016 005301      3$:     DEC      R1     ;DECREMENT ENTRY
2082 003020 006301      ASL      R1         ;SHIFT IT LEFT
2083 003022 016137 100654 003046      MOV      TSTADR(R1),4$ ;GET THE TEST ADDRESS
2084 003030 005237 001300      INC      KYBCTL     ;SET SINGLE TEST INDICATOR
2085 003034 012737 000001 001104      MOV      #1,$ICNT   ;PRESET ITERATION COUNT
2086 003042 000177 000000      JMP      24$        ;GO TO THE SELECTED TEST
2087 003046 000000      4$:     .WORD      0   ;TEST ADDRESS GOES HERE
2088
2089      ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
2090
2091 003050 005737 001302      CHANGE: TST      CHGADR ;CHANGE THE ADDRESS ?
2092 003054 001421      BEQ      3$          ;BR IF NOT
2093 003056 005037 001302      CLR      CHGADR     ;CLEAR THE INDICATOR
2094 003062 104401 071702      1$:     TYPE      ,ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS
2095 003066 013746 001304      MOV      $RMADR,-(SP) ;PUT THE ADDRESS ON THE STACK
2096 003072 104402      TYPOC      ;TYPE THE ACTUAL ADDRESS
2097 003074 104401 001207      TYPE      ,$CRLF    ;CR-LF
2098 003100 104401 071762      TYPE      ,NTRH11   ;ASK FOR NEW ADDRESS
2099 003104 104412      RDOCT
2100 003106 005716      TST      (SP)       ;0 OR 'CR' ENTERED ?
2101 003110 001402      BEQ      2$          ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
2102 003112 011637 001304      MOV      (SP), $RMADR ;NEW RH11 ADDRESS
2103 003116 005726      2$:     TST      (SP)+   ;CORRECT THE STACK POINTER
2104 003120 012737 003140 000004      3$:     MOV      #4,$2#4 ;LOAD TRAP ADDRESS
2105 003126 013700 001304      MOV      $RMADR,R0   ;RH11 ADDRESS
2106 003132 005760 000002      TST      RMWC(R0)   ;SEE IF RH11 RESPONDS AT THAT ADDRESS
2107 003136 000404      BR       5$          ;BR, RH11 ALIVE AT PRESENT ADDRESS
2108 003140 104056      4$:     ERROR    5$    ;NO RESPONSE TO ADDRESS
2109 003142 062706 000004      ADD      #4,SP       ;RESET THE STACK POINTER
2110 003146 000743      BR       1$          ;GET ADDRESS AGAIN
2111 003150 012737 000006 000004      5$:     MOV      #6,$2#4 ;RESTORE THE VECTOR
2112 003156 000207      RTS      PC         ;RETURN
2113
2114      ;;*****
2115      .SBTTL  *** TESTS ***
2116
2117      ;;*****
2118
2119
2120
2121 003160 013700 001304      TST1AA: MOV      $RMADR,R0 ;;RESTORE R0 AFTER END OF PASS
2122
2123      ;;*****
2124      ;*TEST 1      NEUTRAL ACCESS TEST
2125      ;*

```

```

2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136 003164
2137 003164 005737 001300
2138 003170 001406
2139 003172 100002
2140 003174 000137 002676
2141 003200 012737 177777 001300
2142 003206 112737 000001 001102
2143 003214 012737 003236 001106
2144 003222 012737 003236 001110
2145 003230 012737 000001 001176
2146 003236 012706 001100
2147 003242 012760 000040 000010
2148
2149
2150
2151
2152 003250 113760 001224 000010
2153 003256 013737 001224 001240
2154 003264 005760 000012
2155 003270 005037 001250
2156 003274 016037 000010 001126
2157 003302 012737 000010 001122
2158 003310 060037 001122
2159 003314 005037 001124
2160 003320 013737 001126 001164
2161 003326 042737 167777 001164
2162 003334 023737 001124 001164
2163 003342 001414
2164 003344 013737 001126 001174
2165 003352 042737 010000 001174
2166 003360 053737 001174 001124
2167 003366 104037
2168 003370 005137 001250
2169 003374 000240
2170 003376 005737 001250
2171 003402 001403
2172 003404 012760 000040 000010
2173 003412 113760 001226 000010
2174 003420 013737 001226 001240
2175 003426 005760 000012
2176 003432 005037 001250
2177 003436 016037 000010 001126
2178 003444 012737 000010 001122
2179 003452 060037 001122
2180 003456 005037 001124
2181 003462 013737 001126 001164

```

```

; *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 RMD3, THAT THE DRIVE IS ONLINE (RMD51 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 TESTS ?
BEQ 25 ; BR IF NOT
BPL 15 ; BR IF JUST ENTERED TEST
JMP EXEC ; RETURN & GET NEXT TEST NUMBER
15: MOV #1, KYBCTL ; SET SINGLE TEST INDICATOR
25: MOV #1, $STNM ; TEST NUMBER
MOV #TEST1, $LPADR ; LOAD LOOP ON TEST ADDRESS
MOV #TEST1, $LPERR ; LOAD LOOP ON ERROR ADDRESS
MOV #1, $TIMES ; DO 1 ITERATION
TEST1: MOV #STACK, SP ; LOAD THE STACK POINTER
MOV #CLR, RMCS2(R0) ; INITIALIZE THE MASSBUS

```

```

; *****
; VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

```

```

MOV #PORTA, RMCS2(R0) ; SELECT PORT A
MOV #PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RMD51(R0) ; SEE IF DRIVE (PORT A) PRESENT
CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV #RMCS2, $BDDAT ; GET CONTENTS OF RMCS2
MOV #RMCS2, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDAOR ; ADD RHI1 BASE ADDRESS
CLR $GDDAT ; WHAT REGISTER SHOULD BE
MOV $BDDAT, $STMP0 ; MOVE REGISTER CONTENTS TO 'STMP0'
BIC #CNED, $STMP0 ; SAVE SPECIFIED BITS
CMP $GDDAT, $STMP0 ; COMPARE THE BITS
BEQ 645 ; BR IF OK
MOV $BDDAT, $STMP4 ; COPY 'BAD DATA'
BIC #NED, $STMP4 ; CLEAR THE MASKED BITS
BIS $STMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 37 ; TYPE MESSAGE 37
COM CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
645: NOP
TST CKERR ; WAS 'NED' SET ?
BEQ .+10 ; BR IF NOT
MOV #CLR, RMCS2(R0) ; ISSUE MASSBUS INIT TO CLEAR 'NED'
MOV #PORTB, RMCS2(R0) ; SELECT PORT B
MOV #PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RMD51(R0) ; SEE IF DRIVE (PORT B) PRESENT
CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV #RMCS2, $BDDAT ; GET CONTENTS OF RMCS2
MOV #RMCS2, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDAOR ; ADD RHI1 BASE ADDRESS
CLR $GDDAT ; WHAT REGISTER SHOULD BE
MOV $BDDAT, $STMP0 ; MOVE REGISTER CONTENTS TO 'STMP0'

```

F04

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T1 NEUTRAL

MACY11 30(1046) 01-AUG-77 11:02 PAGE 44
ACCESS TEST

SEQ 0046

2182	003470	042737	167777	001164	BIC	#1CNEED,\$TMP0	;SAVE SPECIFIED BITS
2183	003476	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
2184	003504	001414			BEQ	66\$;BR IF OK
2185	003506	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
2186	003514	042737	010000	001174	BIC	#NED,\$TMP4	;CLEAR THE MASKED BITS
2187	003522	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2188	003530	104037			ERROR	37	;TYPE MESSAGE 37
2189	003532	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
2190	003536	000240			NOP		
2191	003540	005737	001250		TST	CKERR	;WAS 'NED' SET ?
2192	003544	001403			BEQ	+.10	;BR IF NOT
2193	003546	012760	000040	000010	MOV	#CLR,RMCS2(R0)	;ISSUE MASSBUS INIT TO CLEAR 'NED'

;CONFIRM THAT DRIVE IS AN RMO3 AND IS DUAL PORT

2198	003554	113760	001224	000010	MOVB	PORTA,RMCS2(R0)	;SELECT PORT A
2199	003562	013737	001224	001240	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2200	003570	005037	001250		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
2201	003574	016037	000026	001126	MOV	RMDT(R0),\$BDDAT	;GET CONTENTS OF RMDT
2202	003602	012737	000026	001122	MOV	#RMDT,\$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
2203	003610	060037	001122		ADD	R0,\$BDAOR	;ADD RHI1 BASE ADDRESS
2204	003614	012737	024024	001124	MOV	#024024,\$GDDAT	;WHAT REGISTER SHOULD BE
2205	003622	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
2206	003630	042737	000003	001164	BIC	#1C177774,\$TMP0	;SAVE SPECIFIED BITS
2207	003636	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
2208	003644	001414			BEQ	68\$;BR IF OK
2209	003646	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
2210	003654	042737	177774	001174	BIC	#177774,\$TMP4	;CLEAR THE MASKED BITS
2211	003662	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2212	003670	104001			ERROR	1	;TYPE MESSAGE 1
2213	003672	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR

2214	003676	000240			NOP		
2215	003700	113760	001226	000010	MOVB	PORTB,RMCS2(R0)	;SELECT PORT B
2216	003706	013737	001226	001240	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2217	003714	005037	001250		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
2218	003720	016037	000026	001126	MOV	RMDT(R0),\$BDDAT	;GET CONTENTS OF RMDT
2219	003726	012737	000026	001122	MOV	#RMDT,\$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
2220	003734	060037	001122		ADD	R0,\$BDAOR	;ADD RHI1 BASE ADDRESS
2221	003740	012737	024024	001124	MOV	#024024,\$GDDAT	;WHAT REGISTER SHOULD BE
2222	003746	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
2223	003754	042737	000003	001164	BIC	#1C177774,\$TMP0	;SAVE SPECIFIED BITS
2224	003762	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
2225	003770	001414			BEQ	70\$;BR IF OK
2226	003772	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
2227	004000	042737	177774	001174	BIC	#177774,\$TMP4	;CLEAR THE MASKED BITS
2228	004006	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2229	004014	104001			ERROR	1	;TYPE MESSAGE 1
2230	004016	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR

70\$:

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

2231	004022	000240			NOP		
2232							
2233							
2234							
2235							
2236	004024	113760	001224	000010	MOVB	PORTA,RMCS2(R0)	;SELECT PORT A
2237	004032	013737	001224	001240	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

2238	004040	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
2239	004044	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
2240	004052	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
2241	004060	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
2242	004064	012737	001000	001124	MOV	#PGM, \$GDDAT	; WHAT REGISTER SHOULD BE
2243	004072	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'STMP0'
2244	004100	042737	176777	001164	BIC	#1CPGM, \$TMP0	; SAVE SPECIFIED BITS
2245	004106	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
2246	004114	001414			BEQ	72S	; BR IF OK
2247	004116	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
2248	004124	042737	001000	001174	BIC	#PGM, \$TMP4	; CLEAR THE MASKED BITS
2249	004132	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2250	004140	104045			ERROR	45	; TYPE MESSAGE 45
2251	004142	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
2252	004146	000240			NOP		
2253	004150	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
2254	004154	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
2255	004162	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
2256	004170	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
2257	004174	012737	010600	001124	MOV	#MOL'DPR'DRY, \$GDDAT	; WHAT REGISTER SHOULD BE
2258	004202	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'STMP0'
2259	004210	042737	167177	001164	BIC	#1C10600, \$TMP0	; SAVE SPECIFIED BITS
2260	004216	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
2261	004224	001414			BEQ	74S	; BR IF OK
2262	004226	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
2263	004234	042737	010600	001174	BIC	#10600, \$TMP4	; CLEAR THE MASKED BITS
2264	004242	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2265	004250	104002			ERROR	2	; TYPE MESSAGE 2
2266	004252	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
2267	004256	000240			NOP		
2268	004260	113760	001226	000010	MOV	PORTB, RMCS2(RO)	; SELECT PORT B
2269	004266	013737	001226	001240	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2270	004274	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
2271	004300	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
2272	004306	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
2273	004314	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
2274	004320	012737	001000	001124	MOV	#PGM, \$GDDAT	; WHAT REGISTER SHOULD BE
2275	004326	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'STMP0'
2276	004334	042737	176777	001164	BIC	#1CPGM, \$TMP0	; SAVE SPECIFIED BITS
2277	004342	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
2278	004350	001414			BEQ	76S	; BR IF OK
2279	004352	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
2280	004360	042737	001000	001174	BIC	#PGM, \$TMP4	; CLEAR THE MASKED BITS
2281	004366	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2282	004374	104045			ERROR	45	; TYPE MESSAGE 45
2283	004376	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
2284	004402	000240			NOP		
2285	004404	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
2286	004410	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
2287	004416	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
2288	004424	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
2289	004430	012737	010600	001124	MOV	#MOL'DPR'DRY, \$GDDAT	; WHAT REGISTER SHOULD BE
2290	004436	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'STMP0'
2291	004444	042737	167177	001164	BIC	#1C10600, \$TMP0	; SAVE SPECIFIED BITS
2292	004452	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
2293	004460	001414			BEQ	78S	; BR IF OK

72S:

74S:

76S:

H04

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T1 NEUTRAL

MACY11 30(1046) 01-AUG-77 11:02 PAGE 46
ACCESS TEST

SEQ 0048

2294	004462	013737	001126	001174	MOV	\$BDDAT, \$STMP4	; COPY 'BAD DATA'
2295	004470	042737	010600	001174	BIC	#10600, \$STMP4	; CLEAR THE MASKED BITS
2296	004476	053737	001174	001124	BIS	\$STMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2297	004504	104002			ERROR	2	; TYPE MESSAGE 2
2298	004506	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
2299	004512	000240			78\$: NOP		

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

2304	004514	113760	001224	000010	MOVB	PORTA, RMCS2(R0)	; SELECT PORT A
2305	004522	016037	000030	001124	MOV	RMSN(R0), \$GDDAT	; STORE THE PORT A SERIAL NUMBER
2306	004530	113760	001226	000010	MOVB	PORTB, RMCS2(R0)	; SELECT PORT B
2307	004536	016037	000030	001126	MOV	RMSN(R0), \$BDDAT	; STORE THE PORT B SERIAL NUMBER
2308	004544	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	; ARE THEY THE SAME ?
2309	004552	001406			BEQ	1\$; BR IF THEY ARE
2310	004554	104003			ERROR	3	; REPORT THE ERROR
2311	004556	032777	100000	174354	BIT	#SW15, @SWR	; HALT ON ERROR ?
2312	004564	001001			BNE	1\$; BR IF SET - PROGRAM HAS ALREADY HALTED
2313	004566	000000			HALT		; HALT, POSSIBLE CABLE CONNECTION PROBLEM
2314	004570	000004			1\$: SCOPE		; LOOP ?

*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.
*

2335	004572				↑ST2:		
2336	004572	005737	001300		TST	KYBCTL	; PERFORMING ONLY SINGLE TESTS ?
2337	004576	001406			BEQ	2\$; BR IF NOT
2338	004600	100002			BPL	1\$; BR IF JUST ENTERED TEST
2339	004602	000137	002676		JMP	EXEC	; RETURN & GET NEXT TEST NUMBER
2340	004606	012737	177777	001300	1\$: MOV	#-1, KYBCTL	; SET SINGLE TEST INDICATOR
2341	004614	112737	000002	001102	2\$: MOVB	#2, \$STNM	; TEST NUMBER
2342	004622	012737	004644	001106	MOV	#TEST2, \$LPADR	; LOAD LOOP ON TEST ADDRESS
2343	004630	012737	004644	001110	MOV	#TEST2, \$LPERR	; LOAD LOOP ON ERROR ADDRESS
2344	004636	012737	000002	001176	MOV	#2, \$TIMES	; DO 2. ITERATIONS
2345	004644	012706	001100		TEST2: MOV	#STACK, SP	; LOAD THE STACK POINTER
2346	004650	012737	000240	177776	MOV	#(5*32.), @#PS	; SET PRIORITY TO 5 IN CASE LOOPING
2347	004656	005037	001262		CLR	TIMEA	; CLEAR TIMEOUT VALUE FOR PORT A
2348	004662	005037	001264		CLR	TIMEAP	; CLEAR UPPER TIMEOUT TOLERANCE
2349	004666	005037	001266		CLR	TIMEAM	; CLEAR LOWER TIMEOUT TOLERANCE


```

2350
2351 ;*****
2352 ;START THE TIMER
2353
2354 004672 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
2355 004676 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
2356
2357 ;*****
2358
2359 ;SEIZE THE DRIVE THROUGH PORT A
2360
2361 004704 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
2362 004712 013737 001224 001242 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
2363 004720 005060 000006 CLR RMDA(RO) ;WRITE RMDA
2364 004724 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
2365 004732 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2366 004740 013737 001226 001244 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
2367 004746 016037 000012 001126 MOV RMDSI(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
2368 004754 010037 001122 RO,$BDAOR ;R#11 BASE ADDRESS
2369 004760 062737 000012 001122 ADD #RMDSI,$BDAOR ;GENERATE BAD REGISTER ADDRESS
2370 004766 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
2371 004772 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
2372 005000 001403 BEQ 64$ ;BR IF IT IS
2373 005002 104004 ERROR 4 ;REPORT THE ERROR
2374 005004 000137 006136 JMP 5$ ;BYPASS REST OF THE SUBTEST
2375 005010
2376 005010 113760 001224 000010 64$: MOVB PORTA,RMCS2(RO) ;SELECT PORT A
2377 005016 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2378 005024 016037 000012 001126 MOV RMDSI(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2379 005032 042737 020001 001126 BIC #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
2380 005040 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2381 005046 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
2382 005054 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
2383 005060 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
2384 005066 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
2385 005074 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
2386 005102 001401 BEQ 65$ ;BR IF THEY ARE
2387 005104 104005 ERROR 5 ;REPORT THE ERROR
2388 005106 000240 65$: NOP
2389
2390 ;*****
2391 ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2392
2393 005110 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
2394 005116 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2395 005124 016046 000046 MOV RMEC2(RO),-(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
2396 005130 016046 000044 MOV RMEC1(RO),-(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
2397 005134 016046 000030 MOV RMSN(RO),-(SP) ;STORE REGISTER RMSN, PORT B, FOR CHECK
2398 005140 016046 000034 MOV RMDC(RO),-(SP) ;STORE REGISTER RMDC, PORT B, FOR CHECK
2399 005144 016046 000032 MOV RMOF(RO),-(SP) ;STORE REGISTER RMOF, PORT B, FOR CHECK
2400 005150 016046 000042 MOV RMER2(RO),-(SP) ;STORE REGISTER RMER2, PORT B, FOR CHECK
2401 005154 016046 000020 MOV RMLA(RO),-(SP) ;STORE REGISTER RMLA, PORT B, FOR CHECK
2402 005160 016046 000026 MOV RMDT(RO),-(SP) ;STORE REGISTER RMDT, PORT B, FOR CHECK
2403 005164 016046 000006 MOV RMDA(RO),-(SP) ;STORE REGISTER RMDA, PORT B, FOR CHECK
2404 005170 016046 000024 MOV RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT B, FOR CHECK
2405 005174 016046 000014 MOV RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT B, FOR CHECK

```

```

2406
2407
2408 ;*****
2409 ;WAIT FOR PORT A TO TIMEOUT
2410 005200 005760 000012 1S: TST RMO51(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
2411 005204 001006 BNE 2S ;BR WHEN TIMEOUT OCCURS
2412 005206 005737 001260 TST WATCH ;CHECK WATCH
2413 005212 001372 BNE 1S ;BR IF NOT ZERO
2414 005214 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
2415 005216 000137 005622 JMP 4S ;BYPASS TIMEOUT TIME CHECK
2416 005222 012737 000340 177776 2S: MOV #(7*32.),2#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2417 005230 013737 001256 001262 MOV TIME,TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
2418 005236 004537 066142 JSR R5,TOLER ;CALCULATE THE TOLERANCE
2419 005242 001262 .WORD TIMEA ;TIMEOUT VALUE FOR PORT A
2420 005244 012637 001264 MOV (SP)+,TIMEAP ;+25% TOLERANCE
2421 005250 012637 001266 MOV (SP)+,TIMEAM ;-25% TOLERANCE
2422
2423 ;*****
2424 ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2425
2426 005254 023727 001256 000764 CMP TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
2427 005262 103001 BHS 3S ;BR IF IT WAS
2428 005264 104055 ERROR 5S ;REPORT TIMEOUT TOO SHORT
2429
2430 ;*****
2431 ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2432
2433 005266 012737 000240 177776 3S: MOV #(5*32.),2#PS ;RESTORE PRIORITY TO 5
2434
2435 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2436
2437 005274 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2438 005300 012737 000012 001122 MOV #RMO51,$B0ADR ;FORM THE ADDRESS OF RMO51 FOR TIMEOUT
2439 005306 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
2440 005312 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2441 005320 113760 001224 000010 MOVVB PORTA, RMCS2(RO) ;SELECT PORT A.
2442 005326 016037 000012 001170 MOV RMO51(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2443 005334 042737 024001 001170 BIC #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
2444 005342 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
2445 005350 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2446 005356 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
2447 005364 016037 000012 001172 MOV RMO51(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2448 005372 042737 024001 001172 BIC #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
2449 005400 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
2450 005406 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2451 005414 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2452 005422 001006 BNE 66S ;BR IF NOT
2453 005424 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2454 005430 001037 BNE 68S ;BR IF NOT
2455 005432 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2456 005434 000137 005620 JMP 70S ;BYPASS THE REST OF THE CHECKS
2457 005440 013737 001170 001126 66S: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2458 005446 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2459 005454 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
2460 005462 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2461 005466 001414 BEQ 67S ;BR IF ZERO

```

K04

```

2462 005470 013737 001224 001240      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2463 005476 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2464 005504 113760 001224 000010      MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
2465 005512 005737 001166      TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2466 005516 001004      BNE     68$            ;BR IF NOT
2467 005520 012737 177777 001154 67$:    MOV      #-1,RELEERR    ;SET 'RELEASE ERROR' INDICATOR
2468 005526 104022      ERROR   22             ;TYPE ERROR MESSAGE 22
2469 005530 013737 001170 001126 68$:    MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
2470 005536 013737 001224 001240      MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
2471 005544 042737 100100 001126      BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
2472 005552 023737 001124 001126      CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
2473 005560 001401      BEQ     69$            ;BR IF OK FROM PORT A.
2474 005562 104007      ERROR   7              ;REPORT ERROR
2475 005564 013737 001172 001126 69$:    MOV      $TMP3,$BDDAT    ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
2476 005572 013737 001226 001240      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
2477 005600 042737 100100 001126      BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
2478 005606 023737 001124 001126      CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
2479 005614 001401      BEQ     70$            ;BR IF OK
2480 005616 104007      ERROR   7              ;REPORT ERROR
2481 005620 000240 70$:    NOP

```

```

2482
2483 ;:*****
2484 ;CHECK THE REGISTERS STORED THROUGH PORT B. ALL REGISTERS SHOULD BE ZERO.
2485 ;THE REGISTERS ARE STORED ON THE STACK.

```

```

2486
2487 005622 013737 001226 001240 4$:    MOV      PORTB,PTNBR    ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2488 005630 010037 001122      MOV      RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMER1
2489 005634 062737 000014 001122      ADD     #RMER1,$BDAOR   ;ADDRESS OF RMER1 FOR TYPEOUT
2490 005642 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMER1
2491 005646 001401      BEQ     .+4             ;CONTENTS ZERO ?
2492 005650 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2493 005652 010037 001122      MOV     RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMMR1
2494 005656 062737 000024 001122      ADD     #RMMR1,$BDAOR  ;ADDRESS OF RMMR1 FOR TYPEOUT
2495 005664 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMMR1
2496 005670 001401      BEQ     .+4             ;CONTENTS ZERO ?
2497 005672 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2498 005674 010037 001122      MOV     RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMDA
2499 005700 062737 000006 001122      ADD     #RMDA,$BDAOR   ;ADDRESS OF RMDA FOR TYPEOUT
2500 005706 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMDA
2501 005712 001401      BEQ     .+4             ;CONTENTS ZERO ?
2502 005714 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2503 005716 010037 001122      MOV     RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMDT
2504 005722 062737 000026 001122      ADD     #RMDT,$BDAOR   ;ADDRESS OF RMDT FOR TYPEOUT
2505 005730 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMDT
2506 005734 001401      BEQ     .+4             ;CONTENTS ZERO ?
2507 005736 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2508 005740 010037 001122      MOV     RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMLA
2509 005744 062737 000020 001122      ADD     #RMLA,$BDAOR   ;ADDRESS OF RMLA FOR TYPEOUT
2510 005752 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMLA
2511 005756 001401      BEQ     .+4             ;CONTENTS ZERO ?
2512 005760 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2513 005762 010037 001122      MOV     RO,$BDAOR      ;BASE ADDRESS FOR REGISTER RMER2
2514 005766 062737 000042 001122      ADD     #RMER2,$BDAOR  ;ADDRESS OF RMER2 FOR TYPEOUT
2515 005774 012637 001126      MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMER2
2516 006000 001401      BEQ     .+4             ;CONTENTS ZERO ?
2517 006002 104006      ERROR   6              ;REPORT THAT PORT B SAW NON-ZERO REGISTER

```

2518	006004	010037	001122	MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RMOF
2519	006010	062737	000032	ADD	#RMOF, \$BDADR	; ADDRESS OF RMOF FOR TYPEOUT
2520	006016	012637	001126	MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RMOF
2521	006022	001401		BEQ	.+4	; CONTENTS ZERO ?
2522	006024	104006		ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2523	006026	010037	001122	MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RMDC
2524	006032	062737	000034	ADD	#RMDC, \$BDADR	; ADDRESS OF RMDC FOR TYPEOUT
2525	006040	012637	001126	MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RMDC
2526	006044	001401		BEQ	.+4	; CONTENTS ZERO ?
2527	006046	104006		ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2528	006050	010037	001122	MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RMSN
2529	006054	062737	000030	ADD	#RMSN, \$BDADR	; ADDRESS OF RMSN FOR TYPEOUT
2530	006062	012637	001126	MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RMSN
2531	006066	001401		BEQ	.+4	; CONTENTS ZERO ?
2532	006070	104006		ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2533	006072	010037	001122	MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RMEC1
2534	006076	062737	000044	ADD	#RMEC1, \$BDADR	; ADDRESS OF RMEC1 FOR TYPEOUT
2535	006104	012637	001126	MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RMEC1
2536	006110	001401		BEQ	.+4	; CONTENTS ZERO ?
2537	006112	104006		ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2538	006114	010037	001122	MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RMEC2
2539	006120	062737	000046	ADD	#RMEC2, \$BDADR	; ADDRESS OF RMEC2 FOR TYPEOUT
2540	006126	012637	001126	MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RMEC2
2541	006132	001401		BEQ	.+4	; CONTENTS ZERO ?
2542	006134	104006		ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2543	006136	000004		SS:	SCOPE	; LOOP ?

```

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

```

2563	006140			TST	KYBCTL	; PERFORMING ONLY SINGLE TESTS ?
2564	006140	005737	001300	BEQ	2\$; BR IF NOT
2565	006144	001406		BPL	1\$; BR IF JUST ENTERED TEST
2566	006146	100002		JMP	EXEC	; RETURN & GET NEXT TEST NUMBER
2567	006150	000137	002676	MOV	#-1, KYBCTL	; SET SINGLE TEST INDICATOR
2568	006154	012737	177777	MOV	#3, \$STSTM	; TEST NUMBER
2569	006162	112737	000003	MOV	#TEST3, \$LPADR	; LOAD LOOP ON TEST ADDRESS
2570	006170	012737	006212	MOV	#TEST3, \$LPERR	; LOAD LOOP ON ERROR ADDRESS
2571	006176	012737	006212	MOV	#2, \$TIMES	; DO 2. ITERATIONS
2572	006204	012737	000002	MOV	#STACK, SP	; LOAD THE STACK POINTER
2573	006212	012706	001100	TEST3:		

```

2574 006216 012737 000240 177776      MOV    #(<5*32.>),a#PS ;SET PRIORITY TO 5 IN CASE LOOPING
2575 006224 005037 001270      CLR    TIMEB           ;CLEAR TIMEOUT VALUE FOR PORT B
2576 006230 005037 001272      CLR    TIMEBP          ;CLEAR UPPER TIMEOUT TOLERANCE
2577 006234 005037 001274      CLR    TIMEBM         ;CLEAR LOWER TIMEOUT TOLERANCE
2578
2579 ;:*****
2580 ;START THE TIMER
2581
2582 006240 005037 001256      CLR    TIME           ;CLEAR THE ELAPSED TIME COUNTER
2583 006244 012737 003720 001260      MOV    #2000.,WATCH  ;SET WATCH TO 2000 MS
2584
2585 ;:*****
2586
2587 ;SEIZE THE DRIVE THROUGH PORT B
2588
2589 006252 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
2590 006260 013737 001226 001242      MOV    PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
2591 006266 005060 000006      CLR    RMDA(RO)      ;WRITE RMDA
2592 006272 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
2593 006300 013737 001224 001240      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2594 006306 013737 001224 001244      MOV    PORTA,OPPR     ;'OPPOSITE' PORT ADDRESS
2595 006314 016037 000012 001126      MOV    RMDSI(RO),SBD0AT ;SEE IF DRIVE SEIZED BY PORT B
2596 006322 010037 001122      MOV    RO,SBDADR      ;R11 BASE ADDRESS
2597 006326 062737 000012 001122      ADD    #RMDSI,SBDADR  ;GENERATE BAD REGISTER ADDRESS
2598 006334 005037 001124      CLR    $GDDAT         ;REGISTER SHOULD BE ZERO
2599 006340 023737 001124 001126      CMP    $GDDAT,$SBDAT  ;IS THE REGISTER ZERO
2600 006346 001403      BEQ    64$           ;BR IF IT IS
2601 006350 104004      ERROR  4             ;REPORT THE ERROR
2602 006352 000137 007504      JMP    5$            ;BYPASS REST OF THE SUBTEST
2603 006356
2604 006356 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
2605 006364 013737 001226 001240      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2606 006372 016037 000012 001126      MOV    RMDSI(RO),SBD0AT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2607 006400 042737 020001 001126      BIC    #OM!PIP,$SBD0AT ;CLEAR DONT CARE BITS
2608 006406 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2609 006414 013737 001124 001166      MOV    $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
2610 006422 005137 001166      COM    $TMP1          ;COMPLEMENT THE EXPECTED STATUS
2611 006426 013737 001126 001164      MOV    $SBDAT,$TMP0   ;SAVE THE ACTUAL STATUS
2612 006434 043737 001166 001164      BIC    $TMP1,$TMP0   ;CLEAR UNWANTED BITS
2613 006442 023737 001124 001164      CMP    $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
2614 006450 001401      BEQ    65$           ;BR IF THEY ARE
2615 006452 104005      ERROR  5             ;REPORT THE ERROR
2616 006454 000240      NOP
2617
2618 ;:*****
2619 ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2620
2621 006456 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
2622 006464 013737 001224 001240      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2623 006472 016046 000046      MOV    RMEC2(RO),-(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
2624 006476 016046 000044      MOV    RMEC1(RO),-(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
2625 006502 016046 000030      MOV    RMSN(RO),-(SP) ;STORE REGISTER RMSN, PORT A, FOR CHECK
2626 006506 016046 000034      MOV    RMOC(RO),-(SP) ;STORE REGISTER RMOC, PORT A, FOR CHECK
2627 006512 016046 000032      MOV    RMOF(RO),-(SP) ;STORE REGISTER RMOF, PORT A, FOR CHECK
2628 006516 016046 000042      MOV    RMER2(RO),-(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
2629 006522 016046 000020      MOV    RMLA(RO),-(SP) ;STORE REGISTER RMLA, PORT A, FOR CHECK

```

```

2630 006526 016046 000026      MOV      RMDT(RO),-(SP) ;STORE REGISTER RMDT, PORT A, FOR CHECK
2631 006532 016046 000006      MOV      RMDA(RO),-(SP) ;STORE REGISTER RMDA, PORT A, FOR CHECK
2632 006536 016046 000024      MOV      RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
2633 006542 016046 000014      MOV      RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK
2634
2635 ;*****
2636 ;WAIT FOR PORT B TO TIMEOUT
2637
2638 006546 005760 000012      1S:     TST      RMDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
2639 006552 001006                BNE      2S ;BR WHEN TIMEOUT OCCURS
2640 006554 005737 001260      TST      WATCH ;CHECK WATCH
2641 006560 001372                BNE      1S ;BR IF NOT ZERO
2642 006562 104036                ERROR    36 ;NO TIMEOUT WITHIN 2 SECONDS
2643 006564 000137 007170      JMP      4S ;BYPASS TIMEOUT TIME CHECK
2644 006570 012737 000340      2S:     MOV      #(<7*32.>),2#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2645 006576 013737 001256      MOV      TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
2646 006604 004537 066142      JSR      RS,TOLER ;CALCULATE THE TOLERANCE
2647 006610 001270                .WORD   TIMEB ;TIMEOUT VALUE FOR PORT B
2648 006612 012637 001272      MOV      (SP)+,TIMEBP ;+25% TOLERANCE
2649 006616 012637 001274      MOV      (SP)+,TIMEBM ;-25% TOLERANCE
2650
2651 ;*****
2652 ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2653
2654 006622 023727 001256      CMP      TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
2655 006630 103001                BHS      3S ;BR IF IT WAS
2656 006632 104055                ERROR    55 ;REPORT TIMEOUT TOO SHORT
2657
2658 ;*****
2659 ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT
2660
2661 006634 012737 000240      3S:     MOV      #(<5*32.>),2#PS ;RESTORE PRIORITY TO 5
2662
2663 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2664
2665 006642 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2666 006646 012737 000012      MOV      #RMDS1,$BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
2667 006654 060037 001122      ADD      RO,$BDDADR ;ADD THE I/O BASE ADDRESS
2668 006660 012737 011600      MOV      #MOL!PGM!DPR!DRY,$SGDDAT ;COMPARISON CONSTANT
2669 006666 113760 001224      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
2670 006674 016037 000012      MOV      RMDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2671 006702 042737 024001      BIC      #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
2672 006710 013737 001170      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
2673 006716 042737 100100      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2674 006724 113760 001226      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
2675 006732 016037 000012      MOV      RMDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2676 006740 042737 024001      BIC      #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
2677 006746 013737 001172      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
2678 006754 042737 100100      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2679 006762 023737 001164      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2680 006770 001006                BNE      66S ;BR IF NOT
2681 006772 005737 001164      TST      STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2682 006776 001037                BNE      68S ;BR IF NOT
2683 007000 104046                ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2684 007002 000137 007166      JMP      70S ;BYPASS THE REST OF THE CHECKS
2685 007006 013737 001170      66S:   MOV      STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE

```

```

2686 007014 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2687 007022 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
2688 007030 005737 001164 TST $TMP0 ;SEE IF STATUS EQ C FROM PORT A.
2689 007034 001414 BEQ 67$ ;BR IF ZERO
2690 007036 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2691 007044 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2692 007052 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
2693 007060 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2694 007064 001004 BNE 68$ ;BR IF NOT
2695 007066 012737 177777 001254 67$: MOV 0-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
2696 007074 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
2697 007076 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDSI READ
2698 007104 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
2699 007112 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
2700 007120 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
2701 007126 001401 BEQ 69$ ;BR IF OK FROM PORT A.
2702 007130 104007 ERROR 7 ;REPORT ERROR
2703 007132 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDSI FOR BIT FAILURES - FROM PORT B.
2704 007140 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2705 007146 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
2706 007154 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
2707 007162 001401 BEQ 70$ ;BR IF OK
2708 007164 104007 ERROR 7 ;REPORT ERROR
2709 007166 000240 70$: NOP
2710
2711 ;*****
2712 ;CHECK THE REGISTERS STORED THROUGH PORT A. ALL REGISTERS SHOULD BE ZERO.
2713 ;THE REGISTERS ARE STORED ON THE STACK.
2714
2715 007170 013737 001224 001240 4$: MOV PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2716 007176 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMR1
2717 007202 062737 000014 001122 ADD #RMR1,$BDADR ;ADDRESS OF RMR1 FOR TYPEOUT
2718 007210 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMR1
2719 007214 001401 BEQ .+4 ;CONTENTS ZERO ?
2720 007216 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2721 007220 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMR1
2722 007224 062737 000024 001122 ADD #RMR1,$BDADR ;ADDRESS OF RMR1 FOR TYPEOUT
2723 007232 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMR1
2724 007236 001401 BEQ .+4 ;CONTENTS ZERO ?
2725 007240 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2726 007242 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
2727 007246 062737 000006 001122 ADD #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
2728 007254 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
2729 007260 001401 BEQ .+4 ;CONTENTS ZERO ?
2730 007262 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2731 007264 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
2732 007270 062737 000026 001122 ADD #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
2733 007276 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
2734 007302 001401 BEQ .+4 ;CONTENTS ZERO ?
2735 007304 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2736 007306 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMLA
2737 007312 062737 000020 001122 ADD #RMLA,$BDADR ;ADDRESS OF RMLA FOR TYPEOUT
2738 007320 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMLA
2739 007324 001401 BEQ .+4 ;CONTENTS ZERO ?
2740 007326 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2741 007330 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMR2
    
```

```

2742 007334 062737 000042 001122 ADD #RMR2,$B0ADR ;ADDRESS OF RMR2 FOR TYPEOUT
2743 007342 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMR2
2744 007346 001401 BEQ .+4 ;CONTENTS ZERO ?
2745 007350 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2746 007352 010037 001122 MOV R0,$B0ADR ;BASE ADDRESS FOR REGISTER RMOF
2747 007356 062737 000032 001122 ADD #RMOF,$B0ADR ;ADDRESS OF RMOF FOR TYPEOUT
2748 007364 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMOF
2749 007370 001401 BEQ .+4 ;CONTENTS ZERO ?
2750 007372 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2751 007374 010037 001122 MOV R0,$B0ADR ;BASE ADDRESS FOR REGISTER RMDC
2752 007400 062737 000034 001122 ADD #RMDC,$B0ADR ;ADDRESS OF RMDC FOR TYPEOUT
2753 007406 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMDC
2754 007412 001401 BEQ .+4 ;CONTENTS ZERO ?
2755 007414 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2756 007416 010037 001122 MOV R0,$B0ADR ;BASE ADDRESS FOR REGISTER RMSN
2757 007422 062737 000030 001122 ADD #RMSN,$B0ADR ;ADDRESS OF RMSN FOR TYPEOUT
2758 007430 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMSN
2759 007434 001401 BEQ .+4 ;CONTENTS ZERO ?
2760 007436 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2761 007440 010037 001122 MOV R0,$B0ADR ;BASE ADDRESS FOR REGISTER RMEC1
2762 007444 062737 000044 001122 ADD #RMEC1,$B0ADR ;ADDRESS OF RMEC1 FOR TYPEOUT
2763 007452 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMEC1
2764 007456 001401 BEQ .+4 ;CONTENTS ZERO ?
2765 007460 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2766 007462 010037 001122 MOV R0,$B0ADR ;BASE ADDRESS FOR REGISTER RMEC2
2767 007466 062737 000046 001122 ADD #RMEC2,$B0ADR ;ADDRESS OF RMEC2 FOR TYPEOUT
2768 007474 012637 001126 MOV (SP)+,$B0DAT ;CHECK THE STORED CONTENTS OF RMEC2
2769 007500 001401 BEQ .+4 ;CONTENTS ZERO ?
2770 007502 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2771 007504 000004 5$: SCOPE ;LOOP ?

```

```

2772
2773 ;*****
2774 ;*TEST 4 PORT 'A' SEIZE/RELEASE TEST
2775 ;*
2776 ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
2777 ;*
2778 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
2779 ;*
2780 ;* B. SET VOLUME VALID AND CLEAR ANY ERROR
2781 ;*
2782 ;* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
2783 ;* RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
2784 ;* DRIVE.
2785 ;*
2786 ;*****

```

```

2787 007506 1ST4:
2788 007506 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2789 007512 001406 BEQ 2$ ;BR IF NOT
2790 007514 100002 BPL 1$ ;BR IF JUST ENTERED TEST
2791 007516 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2792 007522 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2793 007530 112737 000004 001102 2$: MOVB #4,$STNM ;TEST NUMBER
2794 007536 012737 007560 001106 MOV #TEST4,$LPADR ;LOAD LOOP ON TEST ADDRESS
2795 007544 012737 007560 001110 MOV #TEST4,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2796 007552 012737 000031 001176 MOV #25,$TIMES ;DO 25. ITERATIONS
2797 007560 012706 001100 TEST4: MOV #STACK,SP ;LOAD THE STACK POINTER

```



```

2798
2799
2800 ;*****
2801 ;START THE TIMER
2802 007564 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
2803 007570 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
2804
2805 ;*****
2806 ;SEIZE THE DRIVE AND SET VOLUME VALID
2807
2808 ;SEIZE THE DRIVE THROUGH PORT A
2809
2810 007576 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
2811 007604 013737 001224 001242 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
2812 007612 005060 000012 CLR RMD51(RO) ;WRITE RMD51
2813 007616 013737 001226 001244 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
2814 007624 012760 000021 000000 MOV #21, RMCS1(RO) ;SET VOLUME VALID
2815 007632 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2816 007636 016037 000012 001126 MOV RMD51(RO), SBD0AT ;GET CONTENTS OF RMD51
2817 007644 012737 000012 001122 MOV #RMD51, SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2818 007652 060037 001122 ADD RO, SBDADR ;ADD RHL1 BASE ADDRESS
2819 007656 012737 000100 001124 MOV #VV, SGD0AT ;WHAT REGISTER SHOULD BE
2820 007664 013737 001126 001164 MOV SBD0AT, STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2821 007672 042737 177677 001164 BIC #1CVV, STMP0 ;SAVE SPECIFIED BITS
2822 007700 023737 001124 001164 CMP SGD0AT, STMP0 ;COMPARE THE BITS
2823 007706 001414 BEQ 66$ ;BR IF OK
2824 007710 013737 001126 001174 MOV SBD0AT, STMP4 ;COPY 'BAD DATA'
2825 007716 042737 000100 001174 BIC #VV, STMP4 ;CLEAR THE MASKED BITS
2826 007724 053737 001174 001124 BIS STMP4, SGD0AT ;'OR' WITH GOOD DATA FOR TYPEOUT
2827 007732 104013 ERROR 13 ;TYPE MESSAGE 13
2828 007734 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2829 007740 000240 66$: NOP
2830 007742 012760 000040 000010 MOV #CLR, RMCS2(RO) ;CLEAR DRIVE
2831
2832 ;*****
2833
2834 ;RELEASE THE DRIVE FROM PORT A
2835
2836 007750 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
2837 007756 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2838 007764 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
2839
2840 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2841
2842 007772 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2843 007776 012737 000012 001122 MOV #RMD51, SBDADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
2844 010004 060037 001122 ADD RO, SBDADR ;ADD THE I/O BASE ADDRESS
2845 010010 012737 011600 001124 MOV #MOL!PGM!DPR!DRY, SGD0AT ;COMPARISON CONSTANT
2846 010016 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
2847 010024 016037 000012 001170 MOV RMD51(RO), STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2848 010032 042737 024001 001170 BIC #PIP!WRL!OM, STMP2 ;CLEAR DONT CARES
2849 010040 013737 001170 001164 MOV STMP2, STMP0 ;COPY IT INTO 'STMP0'
2850 010046 042737 100100 001164 BIC #ATA!VV, STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2851 010054 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
2852 010062 016037 000012 001172 MOV RMD51(RO), STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2853 010070 042737 024001 001172 BIC #PIP!WRL!OM, STMP3 ;CLEAR DONT CARES

```

E05

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T4

MACY11 30(1046) 01-AUG-77 11:02 PAGE 56
 PORT 'A' SEIZE/RELEASE TEST

SEQ 0058

2854	010076	013737	001172	001166		MOV	\$TMP3,\$TMP1	; COPY IT INTO 'TMP1'
2855	010104	042737	100100	001166		BIC	#ATA!VV,\$TMP1	; CLEAR PORT DEPENDENT BITS FROM THE COPY
2856	010112	023737	001164	001166		CMP	\$TMP0,\$TMP1	; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2857	010120	001006				BNE	68\$; BR IF NOT
2858	010122	005737	001164			TST	\$TMP0	; REGISTERS ARE THE SAME: ARE THEY ZERO ?
2859	010126	001037				BNE	70\$; BR IF NOT
2860	010130	104046				ERROR	46	; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2861	010132	000137	010316			JMP	72\$; BYPASS THE REST OF THE CHECKS
2862	010136	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2863	010144	013737	001226	001240		MOV	PORTB,PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2864	010152	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	; SELECT PORT B.
2865	010160	005737	001164			TST	\$TMP0	; SEE IF STATUS EQ 0 FROM PORT A.
2866	010164	001414				BEQ	69\$; BR IF ZERO
2867	010166	013737	001224	001240		MOV	PORTA,PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2868	010174	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
2869	010202	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	; SELECT PORT A.
2870	010210	005737	001166			TST	\$TMP1	; SEE IF STATUS EQ ZERO FROM PORT B.
2871	010214	001004				BNE	70\$; BR IF NOT
2872	010216	012737	177777	001254	69\$:	MOV	#-1,RELERR	; SET 'RELEASE ERROR' INDICATOR
2873	010224	104022				ERROR	22	; TYPE ERROR MESSAGE 22
2874	010226	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT	; LOOK FOR BIT FAILURES WHEN RMD51 READ
2875	010234	013737	001224	001240		MOV	PORTA,PTNBR	; CHANGE PORT NUMBER
2876	010242	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
2877	010250	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	; ALL BITS OK ?
2878	010256	001401				BEQ	71\$; BR IF OK FROM PORT A.
2879	010260	104007				ERROR	7	; REPORT ERROR
2880	010262	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT	; CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
2881	010270	013737	001226	001240		MOV	PORTB,PTNBR	; CHANGE PORT NUMBER
2882	010276	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
2883	010304	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	; SEE IF READ OK FROM PORT B.
2884	010312	001401				BEQ	72\$; BR IF OK
2885	010314	104007				ERROR	7	; REPORT ERROR
2886	010316	000240			72\$:	NOP		
2887	010320	005737	001254			TST	RELERR	; DID DRIVE RETURN TO NEUTRAL ?
2888	010324	001402				BEQ	46	; BR IF IN NEUTRAL
2889	010326	000137	010602			JMP	1\$; GO WAIT FOR DRIVE TO TIMEOUT
2890	010332	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	; SELECT PORT A
2891	010340	013737	001224	001240		MOV	PORTA,PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2892	010346	005037	001250			CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
2893	010352	016037	000012	001126		MOV	RMD51(RO),\$BDDAT	; GET CONTENTS OF RMD51
2894	010360	012737	000012	001122		MOV	#RMD51,\$BDDAT	; FORM REGISTER ADDRESS OF ERROR MESSAGE
2895	010366	060037	001122			ADD	RO,\$BDDAT	; ADD R11 BASE ADDRESS
2896	010372	005037	001124			CLR	\$GDDAT	; WHAT REGISTER SHOULD BE
2897	010376	013737	001126	001164		MOV	\$BDDAT,\$TMP0	; MOVE REGISTER CONTENTS TO 'TMP0'
2898	010404	042737	077777	001164		BIC	#1CATA,\$TMP0	; SAVE SPECIFIED BITS
2899	010412	023737	001124	001164		CMP	\$GDDAT,\$TMP0	; COMPARE THE BITS
2900	010420	001414				BEQ	73\$; BR IF OK
2901	010422	013737	001126	001174		MOV	\$BDDAT,\$TMP4	; COPY 'BAD DATA'
2902	010430	042737	100000	001174		BIC	#ATA,\$TMP4	; CLEAR THE MASKED BITS
2903	010436	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2904	010444	104017				ERROR	17	; TYPE MESSAGE 17
2905	010446	005137	001250			COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
2906	010452	000240			73\$:	NOP		
2907	010454	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	; SELECT PORT B
2908	010462	013737	001226	001240		MOV	PORTB,PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2909	010470	005037	001250			CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR

F05

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T4

MACY11 30(1046) 01-AUG-77 11:02 PAGE 57
PORT 'A' SEIZE/RELEASE TEST

SEQ 0059

```

2910 010474 016037 000012 001126 MOV RMDS1(RO),SBOADR ;GET CONTENTS OF RMDS1
2911 010502 012737 000012 001122 MOV #RMDS1,SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2912 010510 060037 001122 ADD RO,SBOADR ;ADD RHI1 BASE ADDRESS
2913 010514 005037 001124 CLR SGODAT ;WHAT REGISTER SHOULD BE
2914 010520 013737 001126 001164 MOV SBOADR,$TMPD ;MOVE REGISTER CONTENTS TO 'STMPD'
2915 010526 042737 077777 001164 BIC #1CATA,$TMPD ;SAVE SPECIFIED BITS
2916 010534 023737 001124 001164 CMP SGODAT,$TMPD ;COMPARE THE BITS
2917 010542 001414 BEQ 755 ;BR IF OK
2918 010544 013737 001126 001174 MOV SBOADR,$TMP4 ;COPY 'BAD DATA'
2919 010552 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
2920 010560 053737 001174 001124 BIS $TMP4,$GODAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2921 010566 104017 ERROR 17 ;TYPE MESSAGE 17
2922 010570 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2923 010574 000240 755: NOP
2924 010576 000137 010634 JMP 25 ;GO CHECK FOR LOOP ON ERROR

```

```

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

```

```

2930 010602 15:
2931 010602 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
2932 010610 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2933 010616 005760 000012 TST RMDS1(RO) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
2934 010622 001004 BNE 25 ;BR WHEN DRIVE RELEASED
2935 010624 005737 001260 TST WATCH ;CHECK THE WATCH
2936 010630 001364 BNE 15 ;BR IF NOT ZERO
2937 010632 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
2938 010634 000004 25: SCOPE ;LOOP ?

```

```

*****
;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 RMDS1.
;
; 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.

```

```

2953 *****
2954 TSTS:
2955 010636 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2956 010642 001406 BEQ 25 ;BR IF NOT
2957 010644 100002 BPL 15 ;BR IF JUST ENTERED TEST
2958 010646 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2959 010652 012737 177777 001300 15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2960 010660 112737 000005 001102 25: MOVB #5,$STNM ;TEST NUMBER
2961 010666 012737 010710 001106 MOV #TEST5,$LPADR ;LOAD LOOP ON TEST ADDRESS
2962 010674 012737 010710 001110 MOV #TEST5,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2963 010702 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
2964 010710 012706 001100 TEST5: MOV #STACK,$P ;LOAD THE STACK POINTER
2965

```

```

2966 ;*****
2967 ;START THE TIMER
2968
2969 010714 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
2970 010720 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
2971
2972 ;*****
2973 ;SEIZE THE DRIVE AND SET VOLUME VALID
2974
2975 ;SEIZE THE DRIVE THROUGH PORT B
2976
2977 010726 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
2978 010734 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
2979 010742 005060 000012 CLR RMD51(RO) ;WRITE RMD51
2980 010746 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
2981 010754 012760 000021 000000 MOV #21, RMCS1(RO) ;SET VOLUME VALID
2982 010762 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2983 010766 016037 000012 001126 MOV RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
2984 010774 012737 000012 001122 MOV #RMD51, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2985 011002 060037 001122 ADD RO, $BDDADR ;ADD RHI1 BASE ADDRESS
2986 011006 012737 000100 001124 MOV #VV, $GDDAT ;WHAT REGISTER SHOULD BE
2987 011014 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2988 011022 042737 177677 001164 BIC #ICV, $TMP0 ;SAVE SPECIFIED BITS
2989 011030 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
2990 011036 001414 BEQ 66$ ;BR IF OK
2991 011040 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2992 011046 042737 000100 001174 BIC #VV, $TMP4 ;CLEAR THE MASKED BITS
2993 011054 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2994 011062 104013 ERROR 13 ;TYPE MESSAGE 13
2995 011064 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2996 011070 000240 66$: NOP
2997 011072 012760 000040 000010 MOV #CLR, RMCS2(RO) ;CLEAR DRIVE
2998
2999 ;*****
3000
3001 ;RELEASE THE DRIVE FROM PORT B
3002
3003 011100 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
3004 011106 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3005 011114 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3006
3007 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3008
3009 011122 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3010 011126 012737 000012 001122 MOV #RMD51, $BDDADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
3011 011134 060037 001122 ADD RO, $BDDADR ;ADD THE I/O BASE ADDRESS
3012 011140 012737 011600 001124 MOV #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
3013 011146 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
3014 011154 016037 000012 001170 MOV RMD51(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3015 011162 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
3016 011170 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
3017 011176 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3018 011204 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
3019 011212 016037 000012 001172 MOV RMD51(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3020 011220 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
3021 011226 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'

```

H05

MD-11-DZRMG-A, RM03 DUAL
DZRMGA.P11 01-AUG-77

PORT LOGIC TEST - PART 1
10:58 TS

MACY11 30(1046) 01-AUG-77 11:02 PAGE 59
PORT 'B' SEIZE/RELEASE TEST

SEQ 0061

3022	011234	042737	100100	001166		BIC	#ATA!VV,STMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3023	011242	023737	001164	001166		CMP	STMP0,STMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3024	011250	001006				BNE	68\$;BR IF NOT
3025	011252	005737	001164			TST	STMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3026	011256	001037				BNE	70\$;BR IF NOT
3027	011260	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3028	011262	000137	011446			JMP	72\$;BYPASS THE REST OF THE CHECKS
3029	011266	013737	001170	001126	68\$:	MOV	STMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3030	011274	013737	001226	001240		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3031	011302	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	;SELECT PORT B.
3032	011310	005737	001164			TST	STMP0	;SEE IF STATUS EQ 0 FROM PORT A.
3033	011314	001414				BEQ	69\$;BR IF ZERO
3034	011316	013737	001224	001240		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3035	011324	013737	001172	001126		MOV	STMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3036	011332	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	;SELECT PORT A.
3037	011340	005737	001166			TST	STMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
3038	011344	001004				BNE	70\$;BR IF NOT
3039	011346	012737	177777	001254	69\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
3040	011354	104022				ERROR	22	;TYPE ERROR MESSAGE 22
3041	011356	013737	001170	001126	70\$:	MOV	STMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3042	011364	013737	001224	001240		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
3043	011372	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
3044	011400	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
3045	011406	001401				BEQ	71\$;BR IF OK FROM PORT A.
3046	011410	104007				ERROR	7	;REPORT ERROR
3047	011412	013737	001172	001126	71\$:	MOV	STMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3048	011420	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
3049	011426	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
3050	011434	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
3051	011442	001401				BEQ	72\$;BR IF OK
3052	011444	104007				ERROR	7	;REPORT ERROR
3053	011446	000240			72\$:	NOP		
3054	011450	005737	001254			TST	RELERR	;DID DRIVE RETURN TO NEUTRAL ?
3055	011454	001402				BEQ	+.6	;BR IF IN NEUTRAL
3056	011456	000137	011732			JMP	1\$;GO WAIT FOR DRIVE TO TIMEOUT
3057	011462	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	;SELECT PORT B
3058	011470	013737	001226	001240		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3059	011476	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3060	011502	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
3061	011510	012737	000012	001122		MOV	#RMDS1,\$BDDAT	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3062	011516	060037	001122			ADD	RO,\$BDDAT	;ADD RM11 BASE ADDRESS
3063	011522	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
3064	011526	013737	001126	001164		MOV	\$BDDAT,STMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
3065	011534	042737	077777	001164		BIC	#!CATA,STMP0	;SAVE SPECIFIED BITS
3066	011542	023737	001124	001164		CMP	\$GDDAT,STMP0	;COMPARE THE BITS
3067	011550	001414				BEQ	73\$;BR IF OK
3068	011552	013737	001126	001174		MOV	\$BDDAT,STMP4	;COPY 'BAD DATA'
3069	011560	042737	100000	001174		BIC	#ATA,STMP4	;CLEAR THE MASKED BITS
3070	011566	053737	001174	001124		BIS	STMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3071	011574	104017				ERROR	17	;TYPE MESSAGE 17
3072	011576	005137	001250			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3073	011602	000240			73\$:	NOP		
3074	011604	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	;SELECT PORT A
3075	011612	013737	001224	001240		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3076	011620	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3077	011624	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1

```

3078 011632 012737 000012 001122      MOV      #RMS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3079 011640 060037 001122      ADD      R0, $B0ADR   ;ADD RH11 BASE ADDRESS
3080 011644 005037 001124      CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
3081 011650 013737 001126 001164      MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
3082 011656 042737 077777 001164      BIC      #1CATA, $TMP0 ;SAVE SPECIFIED BITS
3083 011664 023737 001124 001164      CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
3084 011672 001414      BEQ      75$         ;BR IF OK
3085 011674 013737 001126 001174      MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
3086 011702 042737 100000 001174      BIC      #ATA, $TMP4   ;CLEAR THE MASKED BITS
3087 011710 053737 001174 001124      BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3088 011716 104017      ERROR   17          ;TYPE MESSAGE 17
3089 011720 005137 001250      COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
3090 011724 000240      75$:  NOP
3091 011726 000137 011764      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

```

```

3097 011732      1$:  MOV      PORTA, RMCS2(R0) ;SELECT PORT A
3098 011732 113760 001224 000010      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3099 011740 013737 001224 001240      TST      RMS1(R0)    ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3100 011746 005760 000012      BNE      2$         ;BR WHEN DRIVE RELEASED
3101 011752 001004      TST      WATCH      ;CHECK THE WATCH
3102 011754 005737 001260      BNE      1$         ;BR IF NOT ZERO
3103 011760 001364      ERROR   36          ;NO TIMEOUT WITHIN 2 SECONDS
3104 011762 104036      2$:  SCOPE ;LOOP ?
3105 011764 000004

```

```

*****
;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.
;
*****

```

```

3117 011766      1$T6: TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
3118 011766 005737 001300      BEQ      2$         ;BR IF NOT
3119 011772 001406      BPL      1$         ;BR IF JUST ENTERED TEST
3120 011774 100002      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
3121 011776 000137 002676      MOV      #-1, KYBCTL ;SET SINGLE TEST INDICATOR
3122 012002 012737 177777 001300 1$:  MOV      #6, $STNM   ;TEST NUMBER
3123 012010 112737 000006 001102 2$:  MOV      $TEST6, $LPAOR ;LOAD LOOP ON TEST ADDRESS
3124 012016 012737 012040 001106      MOV      $TEST6, $LPERR ;LOAD LOOP ON ERROR ADDRESS
3125 012024 012737 012040 001110      MOV      #25, $TIMES ;DO 25. ITERATIONS
3126 012032 012737 000031 001176      MOV      $STACK, SP ;LOAD THE STACK POINTER
3127 012040 012706 001100      TEST6: MOV      PORTA, RMCS2(R0) ;SELECT PORT A
3128 012044 113760 001224 000010      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3129 012052 013737 001224 001240      MOV      PORTA, SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3130 012060 013737 001224 001242

```

```

*****
;ISSUE A RELEASE COMMAND

```

```

3131
3132
3133

```

```

3134
3135 012066 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE A RELEASE COMMAND
3136
3137 ;*****
3138 ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
3139
3140
3141 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3142
3143 012074 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3144 012100 012737 000012 001122      MOV      #RMDS1,$B0ADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3145 012106 060037 001122      ADD     RO,$B0ADR ;ADD THE I/O BASE ADDRESS
3146 012112 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3147 012120 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
3148 012126 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3149 012134 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
3150 012142 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
3151 012150 042737 100100 001164      BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3152 012156 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
3153 012164 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3154 012172 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
3155 012200 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
3156 012206 042737 100100 001166      BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3157 012214 023737 001164 001166      CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3158 012222 001006      BNE     64$ ;BR IF NOT
3159 012224 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3160 012230 001045      BNE     66$ ;BR IF NOT
3161 012232 104046      ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3162 012234 000137 012434      JMP     68$ ;BYPASS THE REST OF THE CHECKS
3163 012240 013737 001170 001126 64$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3164 012246 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3165 012254 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
3166 012262 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3167 012266 001414      BEQ     65$ ;BR IF ZERO
3168 012270 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3169 012276 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3170 012304 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
3171 012312 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3172 012316 001012      BNE     66$ ;BR IF NOT
3173 012320 012737 177777 001254 65$: MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3174 012326 012760 000011 000000      MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
3175 012334 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
3176 012342 104030      ERROR   30 ;TYPE ERROR MESSAGE 30
3177 012344 013737 001170 001126 66$: MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3178 012352 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
3179 012360 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3180 012366 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
3181 012374 001401      BEQ     67$ ;BR IF OK FROM PORT A.
3182 012376 104007      ERROR   7 ;REPORT ERROR
3183 012400 013737 001172 001126 67$: MOV      $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3184 012406 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
3185 012414 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3186 012422 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
3187 012430 001401      BEQ     68$ ;BR IF OK
3188 012432 104007      ERROR   7 ;REPORT ERROR
3189 012434 000240      NOP

```

```

3190 012436 000004
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201 012440
3202 012440 005737 001300
3203 012444 001406
3204 012446 100002
3205 012450 000137 002676
3206 012454 012737 177777 001300
3207 012462 112737 000007 001102
3208 012470 012737 012512 001106
3209 012476 012737 012512 001110
3210 012504 012737 000031 001176
3211 012512 012706 001100
3212 012516 113760 001226 000010
3213 012524 013737 001226 001240
3214 012532 013737 001226 001242
3215
3216
3217
3218
3219 012540 012760 000013 000000
3220
3221
3222
3223
3224
3225
3226
3227 012546 005037 001254
3228 012552 012737 000012 001122
3229 012560 060037 001122
3230 012564 012737 011700 001124
3231 012572 113760 001224 000010
3232 012600 016037 000012 001170
3233 012606 042737 024001 001170
3234 012614 013737 001170 001164
3235 012622 042737 100100 001164
3236 012630 113760 001226 000010
3237 012636 016037 000012 001172
3238 012644 042737 024001 001172
3239 012652 013737 001172 001166
3240 012660 042737 100100 001166
3241 012666 023737 001164 001166
3242 012674 001006
3243 012676 005737 001164
3244 012702 001045
3245 012704 104046

```

```

SCOPE ;LOOP ?
;*****
*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.
*
;*****
TEST7:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #7,$STNM ;TEST NUMBER
MOV #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST7: MOV #STACK,SP ;LOAD THE STACK POINTER
MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
;*****
;ISSUE A RELEASE COMMAND
MOV #13,RMCS1(RO) ;ISSUE A RELEASE COMMAND
;*****
;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RMDS1,$BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,$GDAT ;COMPARISON CONSTANT
MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
MOV RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
BIC #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
MOV RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
BIC #PIP!WAL!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 645 ;BR IF NOT
TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 665 ;BR IF NOT
ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED

```



```

3246 012706 000137 013106          JMP      68$          ; BYPASS THE REST OF THE CHECKS
3247 012712 013737 001170 001126 64$:  MOV     $TMP2,$BDDAT ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3248 012720 013737 001226 001240          MOV     PORTB,PTNBR  ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3249 012726 113760 001226 000010          MOVB   PORTB, RMCS2(RO) ; SELECT PORT B.
3250 012734 005737 001164          TST    $TMP0        ; SEE IF STATUS EQ 0 FROM PORT A.
3251 012740 001414          BEQ    65$          ; BR IF ZERO
3252 012742 013737 001224 001240          MOV     PORTA,PTNBR  ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3253 012750 013737 001172 001126          MOV     $TMP3,$BDDAT ; 'BAD DATA' FOR ERROR TYPE OUT
3254 012756 113760 001224 000010          MOVB   PORTA, RMCS2(RO) ; SELECT PORT A.
3255 012764 005737 001164          TST    $TMP1        ; SEE IF STATUS EQ ZERO FROM PORT B.
3256 012770 001012          BNE    66$          ; BR IF NOT
3257 012772 012737 177777 001254 65$:  MOV     #-1,RELEERR ; SET 'RELEASE ERROR' INDICATOR
3258 013000 012760 000011 000000          MOV     #11, RMCS1(RO) ; CLEAR THE DRIVE
3259 013006 012760 000013 000000          MOV     #13, RMCS1(RO) ; RELEASE THE DRIVE
3260 013014 104030          ERROR  30          ; TYPE ERROR MESSAGE 30
3261 013016 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ; LOOK FOR BIT FAILURES WHEN RMD51 READ
3262 013024 013737 001224 001240          MOV     PORTA,PTNBR  ; CHANGE PORT NUMBER
3263 013032 042737 100000 001126          BIC     #ATA,$BDDAT  ; DON'T CHECK THE ATTN BIT
3264 013040 023737 001124 001126          CMP     $GDDAT,$BDDAT ; ALL BITS OK ?
3265 013046 001401          BEQ    67$          ; BR IF OK FROM PORT A.
3266 013050 104007          ERROR  7          ; REPORT ERROR
3267 013052 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT ; CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
3268 013060 013737 001226 001240          MOV     PORTB,PTNBR  ; CHANGE PORT NUMBER
3269 013066 042737 100000 001126          BIC     #ATA,$BDDAT  ; DON'T CHECK THE ATTN BIT
3270 013074 023737 001124 001126          CMP     $GDDAT,$BDDAT ; SEE IF READ OK FROM PORT B.
3271 013102 001401          BEQ    68$          ; BR IF OK
3272 013104 104007          ERROR  7          ; REPORT ERROR
3273 013106 000240          NOP
3274 013110 000004          SCOPE          ; LOOP ?

```

```

3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296 013112
3297 013112 005737 001300
3298 013116 001406
3299 013120 100002
3300 013122 000137 002676
3301 013126 012737 177777 001300 1$:

```

*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 RMD51.
*
* 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.
*

```

†ST10:  TST     KYBCTL          ; PERFORMING ONLY SINGLE TESTS ?
        BEQ    2$          ; BR IF NOT
        BPL   1$          ; BR IF JUST ENTERED TEST
        JMP   EXEC        ; RETURN & GET NEXT TEST NUMBER
        MOV   #-1,KYBCTL  ; SET SINGLE TEST INDICATOR

```

M05

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T10

MACY11 30(1046) 01-AUG-77 11:02 PAGE 64
 PORT 'A' RELEASE INTERFERENCE TEST

SEQ 0066

```

3302 013134 112737 000010 001102 25:  MOVB  #10,$STNM      ;TEST NUMBER
3303 013142 012737 013164 001106      MOV  #TEST10,$LPADR ;LOAD LOOP ON TEST ADDRESS
3304 013150 012737 013164 001110      MOV  #TEST10,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3305 013156 012737 000031 001176      MOV  #25,$TIMES    ;DO 25. ITERATIONS
3306 013164 012706 001100      TEST10: MOV  #STACK,$SP ;LOAD THE STACK POINTER
3307
3308                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
3309
3310 013170 113760 001224 000010      MOVB  PORTA,$RMCS2($RO) ;SELECT PORT #A
3311 013176 005060 000012      CLR   $RMS1($RO)      ;SEIZE THE DRIVE
3312 013202 012760 000011 000000      MOV  #11,$RMS1($RO)   ;ISSUE DRIVE CLEAR
3313 013210 012760 000013 000000      MOV  #13,$RMS1($RO)   ;RELEASE THE DRIVE
3314 013216 113760 001226 000010      MOVB  PORTB,$RMCS2($RO) ;SELECT PORT #B
3315 013224 005060 000012      CLR   $RMS1($RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
3316 013230 012760 000011 000000      MOV  #11,$RMS1($RO)   ;ISSUE DRIVE CLEAR
3317 013236 012760 000013 000000      MOV  #13,$RMS1($RO)   ;RELEASE THE DRIVE
3318
3319                                     ;*****
3320
3321                                     ;SEIZE THE DRIVE THROUGH PORT B
3322
3323 013244 113760 001226 000010      MOVB  PORTB,$RMCS2($RO) ;SELECT PORT B
3324 013252 013737 001226 001242      MOV  PORTB,$SEIZPT    ;STORE SEIZING PORT'S ADDRESS
3325 013260 005060 000012      CLR   $RMS1($RO)      ;WRITE RMS1
3326 013264 113760 001224 000010      MOVB  PORTA,$RMCS2($RO) ;SELECT PORT A
3327 013272 013737 001224 001240      MOV  PORTA,$PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3328 013300 013737 001224 001244      MOV  PORTA,$OPPRT     ;'OPPOSITE' PORT ADDRESS
3329 013306 016037 000012 001126      MOV  $RMS1($RO),$SBDAT ;SEE IF DRIVE SEIZED BY PORT B
3330 013314 010037 001122      MOV  $RO,$SBDADR      ;R#11 BASE ADDRESS
3331 013320 062737 000012 001122      ADD  #RMS1,$SBDADR    ;GENERATE BAD REGISTER ADDRESS
3332 013326 005037 001124      CLR   $GDDAT          ;REGISTER SHOULD BE ZERO
3333 013332 023737 001124 001126      CMP  $GDDAT,$SBDAT    ;IS THE REGISTER ZERO
3334 013340 001403      BEQ  645              ;BR IF IT IS
3335 013342 104004      ERROR 4               ;REPORT THE ERROR
3336 013344 000137 014354      JMP  15               ;BYPASS REST OF THE SUBTEST
3337 013350
3338 013350 113760 001226 000010      645: MOVB  PORTB,$RMCS2($RO) ;SELECT PORT B
3339 013356 013737 001226 001240      MOV  PORTB,$PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3340 013364 016037 000012 001126      MOV  $RMS1($RO),$SBDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
3341 013372 042737 020001 001126      BIC  #OM!PIP,$SBDAT   ;CLEAR DONT CARE BITS
3342 013400 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
3343 013406 013737 001124 001166      MOV  $GDDAT,$STMP1    ;USE GOOD DATA AS A MASK
3344 013414 005137 001166      COM  $STMP1           ;COMPLEMENT THE EXPECTED STATUS
3345 013420 013737 001126 001164      MOV  $SBDAT,$STMP0    ;SAVE THE ACTUAL STATUS
3346 013426 043737 001166 001164      BIC  $STMP1,$STMP0    ;CLEAR UNWANTED BITS
3347 013434 023737 001124 001164      CMP  $GDDAT,$STMP0    ;ARE THE EXPECTED STATUS BITS SET ?
3348 013442 001401      BEQ  655              ;BR IF THEY ARE
3349 013444 104005      ERROR 5               ;REPORT THE ERROR
3350 013446 000240      655: NOP
3351
3352                                     ;*****
3353                                     ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
3354
3355 013450 113760 001224 000010      MOVB  PORTA,$RMCS2($RO) ;SELECT PORT A
3356 013456 013737 001224 001240      MOV  PORTA,$PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3357 013464 012760 000013 000000      MOV  #13,$RMS1($RO)   ;ISSUE A RELEASE COMMAND THROUGH PORT A
    
```

```

3358
3359
3360
3361
3362 013472 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
3363 013476 016037 000012 001126    MOV      RMD51(SBDDAT) ;GET CONTENTS OF RMD51
3364 013504 012737 000012 001122    MOV      #RMD51,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3365 013512 060037 001122          ADD      RD,SBADR      ;ADD RHI1 BASE ADDRESS
3366 013516 005037 001124          CLR      $GDCAT       ;WHAT REGISTER SHOULD BE
3367 013522 023737 001124 001126    CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
3368 013530 001403          BEQ      66$          ;BR IF OK
3369 013532 104010          ERROR    10          ;REPORT THE ERROR
3370 013534 005137 001250          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
3371 013540 016037 000000 001126 66$:  MOV      RMCS1(RD),SBDDAT ;GET THE CONTENTS OF RMCS1
3372 013546 012737 000000 001122    MOV      #RMCS1,SBADR  ;FORM ADDRESS OF REGISTER
3373 013554 060037 001122          ADD      RD,SBADR      ;ADDRESS BASE
3374 013560 032737 020000 001126    BIT      #MCPE,SBDDAT  ;IS 'MCPE' SET ?
3375 013566 001404          BEQ      67$          ;BR IF NOT
3376 013570 104011          ERROR    11          ;REPORT THE ERROR
3377 013572 012760 040000 000000    MOV      #TRE,RMCS1(RD) ;CLEAR 'MCPE'
3378 013600 001240          67$:  NOP
3379 013602 005737 001250          TST      CKERR        ;WAS RMD51 NON ZERO ?
3380 013606 001402          BEQ      .+6          ;CONTENTS OF RMD51 SEEN BY PORT A
3381 013610 000137 014354          JMP      1$          ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
3382
3383
3384
3385
3386
3387 013614 113760 001226 000010    MOV      PORTB,RMCS2(RD) ;SELECT PORT B
3388 013622 013737 001226 001240    MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3389 013630 012760 000013 000000    MOV      #13,RMCS1(RD) ;ISSUE RELEASE THROUGH PORT B
3390
3391
3392
3393 013636 005037 001254          CLR      RELERR       ;CLEAR 'RELEASE ERROR' INDICATOR
3394 013642 012737 111700 001124    MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3395 013650 012737 000012 001122    MOV      #RMD51,SBADR  ;REGISTER ADDRESS INCREMENT
3396 013656 060037 001122          ADD      RD,SBADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
3397 013662 113760 001224 000010    MOV      PORTA,RMCS2(RD) ;SELECT PORT A
3398 013670 013737 001224 001240    MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3399 013676 016037 000012 001164    MOV      RMD51(RD),$TMPD ;READ STATUS REGISTER FROM PORT A
3400 013704 113760 001226 000010    MOV      PORTB,RMCS2(RD) ;SELECT PORT B
3401 013712 013737 001226 001240    MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3402 013720 016037 000012 001126    MOV      RMD51(RD),SBDDAT ;DRIVE STATUS FROM PORT B
3403 013726 001404          BEQ      68$          ;BR IF STATUS FROM PORT B ZERO
3404 013730 005737 001164          TST      $TMPD        ;IS STATUS FROM PORT A ZERO ?
3405 013734 001401          BEQ      68$          ;BR IF ZERO
3406 013736 104031          ERROR    31          ;REPORT DRIVE IN NEUTRAL
3407 013740 013737 001164 001126 68$:  MOV      $TMPD,SBDDAT  ;CHECK STATUS FROM PORT A
3408 013746 013737 001224 001240    MOV      PORTA,PTNBR  ;CHANGE PORT ADDRESS FOR TYPEOUT
3409 013754 023737 001124 001126    CMP      $GDDAT,SBDDAT ;COMPARE WITH CONSTANT
3410 013762 001401          BEQ      69$          ;BR IF OK
3411 013764 104027          ERROR    27          ;REPORT REGISTER ERROR
3412 013766 000240          69$:  NOP
3413

```

```

3414 ;RELEASE THE DRIVE FROM PORT A
3415
3416 013770 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
3417 013776 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3418 014004 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
3419
3420 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3421
3422 014012 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3423 014016 012737 000012 001122 MOV #RMS1, $BDDADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
3424 014024 060037 001122 ADD RO, $BDDADR ;ADD THE I/O BASE ADDRESS
3425 014030 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3426 014036 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
3427 014044 016037 000012 001170 MOV RMS1(RO), STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3428 014052 042737 024001 001170 BIC #PIP!WRL!OM, STMP2 ;CLEAR DONT CARES
3429 014060 013737 001170 001164 MOV STMP2, STMP0 ;COPY IT INTO 'STMP0'
3430 014066 042737 100100 001164 BIC #ATA!VV, STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3431 014074 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
3432 014102 016037 000012 001172 MOV RMS1(RO), STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3433 014110 042737 024001 001172 BIC #PIP!WRL!OM, STMP3 ;CLEAR DONT CARES
3434 014116 013737 001172 001166 MOV STMP3, STMP1 ;COPY IT INTO 'STMP1'
3435 014124 042737 100100 001166 BIC #ATA!VV, STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3436 014132 023737 001164 001166 CMP STMP0, STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3437 014140 001006 BNE 70S ;BR IF NOT
3438 014142 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3439 014146 001045 BNE 72S ;BR IF NOT
3440 014150 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3441 014152 000137 014352 JMP 74S ;BYPASS THE REST OF THE CHECKS
3442 014156 013737 001170 001126 70S: MOV STMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3443 014164 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3444 014172 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
3445 014200 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3446 014204 001414 BEQ 71S ;BR IF ZERO
3447 014206 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3448 014214 013737 001172 001126 MOV STMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3449 014222 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
3450 014230 005737 001164 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3451 014234 001012 BNE 72S ;BR IF NOT
3452 014236 012737 177777 001254 71S: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
3453 014244 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
3454 014252 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
3455 014260 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
3456 014262 013737 001170 001126 72S: MOV STMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
3457 014270 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
3458 014276 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
3459 014304 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
3460 014312 001401 BEQ 73S ;BR IF OK FROM PORT A.
3461 014314 104007 ERROR 7 ;REPORT ERROR
3462 014316 013737 001172 001126 73S: MOV STMP3, $BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
3463 014324 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
3464 014332 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
3465 014340 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
3466 014346 001401 BEQ 74S ;BR IF OK
3467 014350 104007 ERROR 7 ;REPORT ERROR
3468 014352 000240 74S: NOP
3469 014354 000004 1S: SCOPE ;LOOP ?

```

3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490 014356
3491 014356 005737 001300
3492 014362 001406
3493 014364 100002
3494 014366 000137 002676
3495 014372 012737 177777 001300
3496 014400 112737 000011 001102
3497 014406 012737 014430 001106
3498 014414 012737 014430 001110
3499 014422 012737 000031 001176
3500 014430 012706 001100
3501
3502
3503
3504 014434 113760 001224 000010
3505 014442 005060 000012
3506 014446 012760 000011 000000
3507 014454 012760 000013 000000
3508 014462 113760 001226 000010
3509 014470 005060 000012
3510 014474 012760 000011 000000
3511 014502 012760 000013 000000
3512
3513
3514
3515
3516
3517 014510 113760 001224 000010
3518 014516 013737 001224 001242
3519 014524 005060 000012
3520 014530 113760 001226 000010
3521 014536 013737 001226 001240
3522 014544 013737 001226 001244
3523 014552 016037 000012 001126
3524 014560 010037 001122
3525 014564 062737 000012 001122

```

*****
*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 RMD51.
*
*  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.
*****
†ST11:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      25          ;BR IF NOT
      BPL      15          ;BR IF JUST ENTERED TEST
      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
15:   MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
25:   MOVB     #11,$STNM   ;TEST NUMBER
      MOV      #TEST11,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST11,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #25,$TIMES  ;DO 25. ITERATIONS
TEST11: MOV     #STACK,$SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
      MOVB     PORTA,RMCS2(RO) ;SELECT PORT #A
      CLR      RMD51(RO)      ;SEIZE THE DRIVE
      MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
      MOVB     PORTB,RMCS2(RO) ;SELECT PORT #B
      CLR      RMD51(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
      MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE

;*****
;SEIZE THE DRIVE THROUGH PORT A
      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
      CLR      RMD51(RO)      ;WRITE RMD51
      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
      MOV      RMD51(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
      MOV      RO,$BDAADR ;R#11 BASE ADDRESS
      ADD      #RMD51,$BDAADR ;GENERATE BAD REGISTER ADDRESS

```

```

3526 014572 005037 001124 CLR $GDDAT ; REGISTER SHOULD BE ZERO
3527 014576 023737 001124 001126 CMP $GDDAT,$BDDAT ; IS THE REGISTER ZERO
3528 014604 001403 BEQ 64$ ; BR IF IT IS
3529 014606 104004 ERROR 4 ; REPORT THE ERROR
3530 014610 000137 015620 JMP 1$ ; BYPASS REST OF THE SUBTEST
3531 014614 64$:
3532 014614 113760 001224 000010 MOVB PORTA, RMCS2(RO) ; SELECT PORT A
3533 014622 013737 001224 001240 MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3534 014630 016037 000012 001126 MOV RMDS1(RO), $BDDAT ; SEE IF SEIZING PORT SEES CORRECT STATUS
3535 014636 042737 020001 001126 BIC #OM!PIP,$BDDAT ; CLEAR DONT CARE BITS
3536 014644 012737 011700 001124 MOV #MOL!FGM!DPR!DRY!VV,$GDDAT ; EXPECTED STATUS
3537 014652 013737 001124 001166 MOV $GDDAT,$STMP1 ; USE GOOD DATA AS A MASK
3538 014660 005137 001166 COM $STMP1 ; COMPLEMENT THE EXPECTED STATUS
3539 014664 013737 001126 001164 MOV $BDDAT,$STMP0 ; SAVE THE ACTUAL STATUS
3540 014672 043737 001166 001164 BIC $STMP1,$STMP0 ; CLEAR UNWANTED BITS
3541 014700 023737 001124 001164 CMP $GDDAT,$STMP0 ; ARE THE EXPECTED STATUS BITS SET ?
3542 014706 001401 BEQ 65$ ; BR IF THEY ARE
3543 014710 104005 ERROR 5 ; REPORT THE ERROR
3544 014712 000240 65$: NOP
3545
3546 ;*****
3547 ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
3548
3549 014714 113760 001226 000010 MOVB PORTB, RMCS2(RO) ; SELECT PORT B
3550 014722 013737 001226 001240 MOV PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3551 014730 012760 000013 000000 MOV #13, RMCS1(RO) ; ISSUE A RELEASE COMMAND THROUGH PORT B
3552
3553 ;*****
3554 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
3555
3556 014736 005037 001250 CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
3557 014742 016037 000012 001126 MOV RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
3558 014750 012737 000012 001122 MOV #RMDS1,$BDDADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
3559 014756 060037 001122 ADD RO,$BDDADR ; ADD RH11 BASE ADDRESS
3560 014762 005037 001124 CLR $GDDAT ; WHAT REGISTER SHOULD BE
3561 014766 023737 001124 001126 CMP $GDDAT,$BDDAT ; IS THE REGISTER OK ?
3562 014774 001403 BEQ 66$ ; BR IF OK
3563 014776 104010 ERROR 10 ; REPORT THE ERROR
3564 015000 005137 001250 COM CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
3565 015004 016037 000000 001126 66$: MOV RMCS1(RO), $BDDAT ; GET THE CONTENTS OF RHCS1
3566 015012 012737 000000 001122 MOV #RMCS1,$BDDADR ; FORM ADDRESS OF REGISTER
3567 015020 060037 001122 ADD RO,$BDDADR ; ADDRESS BASE
3568 015024 032737 020000 001126 BIT #MCPE,$BDDAT ; IS 'MCPE' SET ?
3569 015032 001404 BEQ 67$ ; BR IF NOT
3570 015034 104011 ERROR 11 ; REPORT THE ERROR
3571 015036 012760 040000 000000 MOV #TRE, RMCS1(RO) ; CLEAR 'MCPE'
3572 015044 000240 67$: NOP
3573 015046 005737 001250 TST CKERR ; WAS RMDS1 NON ZERO ?
3574 015052 001402 BEQ .+6 ; CONTENTS OF RMDS1 SEEN BY PORT B
3575 015054 000137 015620 JMP 1$ ; DRIVE IN NEUTRAL, BYPASS REST OF TEST
3576
3577 ;*****
3578 ;RELEASE THE DRIVE FROM PORT A
3579
3580
3581 015060 113760 001224 000010 MOVB PORTA, RMCS2(RO) ; SELECT PORT A

```

E06

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T11

MACY11 30(1046) 01-AUG-77 11:02 PAGE 69
 PORT 'B' RELEASE INTERFERENCE TEST

SEQ 0071

```

3582 015066 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3583 015074 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
3584
3585 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
3586
3587 015102 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
3588 015106 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3589 015114 012737 000012 001122 MOV #RMS1,$BODADR ;REGISTER ADDRESS INCREMENT
3590 015122 060037 001122 ADD RO,$BODADR ;REGISTER BASE ADDRESS FOR TYPEOUT
3591 015126 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
3592 015134 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3593 015142 016037 000012 001164 MOV RMS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
3594 015150 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
3595 015156 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3596 015164 016037 000012 001126 MOV RMS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
3597 015172 001404 BEQ 68$ ;BR IF STATUS FROM PORT A ZERO
3598 015174 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO ?
3599 015200 001401 BEQ 68$ ;BR IF ZERO
3600 015202 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
3601 015204 013737 001164 001126 68$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
3602 015212 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
3603 015220 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
3604 015226 001401 BEQ 69$ ;BR IF OK
3605 015230 104027 ERROR 27 ;REPORT REGISTER ERROR
3606 015232 000240 69$: NOP
3607
3608 ;RELEASE THE DRIVE FROM PORT B
3609
3610 015234 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
3611 015242 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3612 015250 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3613
3614 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3615
3616 015256 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3617 015262 012737 000012 001122 MOV #RMS1,$BODADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
3618 015270 060037 001122 ADD RO,$BODADR ;ADD THE I/O BASE ADDRESS
3619 015274 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3620 015302 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
3621 015310 016037 000012 001170 MOV RMS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3622 015316 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
3623 015324 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
3624 015332 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3625 015340 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
3626 015346 016037 000012 001172 MOV RMS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3627 015354 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
3628 015362 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
3629 015370 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3630 015376 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3631 015404 001006 BNE 70$ ;BR IF NOT
3632 015406 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3633 015412 001045 BNE 72$ ;BR IF NOT
3634 015414 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3635 015416 000137 015616 JMP 74$ ;BYPASS THE REST OF THE CHECKS
3636 015422 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3637 015430 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
  
```

```

3638 015436 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ; SELECT PORT B.
3639 015444 005737 001164              TST    $TMPD           ; SEE IF STATUS EQ 0 FROM PORT A.
3640 015450 001414              BEQ    71$            ; BR IF ZERO
3641 015452 013737 001224 001240      MOV    PORTA, PTNBR   ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3642 015460 013737 001172 001126      MOV    $TMP3, $BDDAT ; 'BAD DATA' FOR ERROR TYPE OUT
3643 015466 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ; SELECT PORT A.
3644 015474 005737 001166              TST    $TMP1         ; SEE IF STATUS EQ ZERO FROM PORT B.
3645 015500 001012              BNE    72$          ; BR IF NOT
3646 015502 012737 177777 001254 71$:  MOV    #-1, RELERR   ; SET 'RELEASE ERROR' INDICATOR
3647 015510 012760 000011 000000      MOV    #11, RMCS1(RO) ; CLEAR THE DRIVE
3648 015516 012760 000013 000000      MOV    #13, RMCS1(RO) ; RELEASE THE DRIVE
3649 015524 104026              ERROR  26           ; TYPE ERROR MESSAGE 26
3650 015536 013737 001170 001126 72$:  MOV    $TMP2, $BDDAT ; LOOK FOR BIT FAILURES WHEN RMD51 READ
3651 015534 013737 001224 001240      MOV    PORTA, PTNBR  ; CHANGE PORT NUMBER
3652 015542 042737 100000 001126      BIC    #ATA, $BDDAT  ; DON'T CHECK THE ATTN BIT
3653 015550 023737 001124 001126      CMP    $GDDAT, $BDDAT ; ALL BITS OK ?
3654 015556 001401              BEQ    73$          ; BR IF OK FROM PORT A.
3655 015560 104007              ERROR  7           ; REPORT ERROR
3656 015562 013737 001172 001126 73$:  MOV    $TMP3, $BDDAT ; CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
3657 015570 013737 001226 001240      MOV    PORTB, PTNBR ; CHANGE PORT NUMBER
3658 015576 042737 100000 001126      BIC    #ATA, $BDDAT  ; DON'T CHECK THE ATTN BIT
3659 015604 023737 001124 001126      CMP    $GDDAT, $BDDAT ; SEE IF READ OK FROM PORT C.
3660 015612 001401              BEQ    74$          ; BR IF OK
3661 015614 104007              ERROR  7           ; REPORT ERROR
3662 015616 000240 74$:  NOP
3663 015620 000004 1$:  SCOPE ; LOOP ?

```

```

3664
3665
3666 ;*****
3667 ;TEST 12      PORT 'A' RELEASE W/ERRORS TEST
3668 ;
3669 ;VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
3670 ;      BITS ARE SET IN THE DRIVE.
3671 ;
3672 ; A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
3673 ;
3674 ; B. WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
3675 ;
3676 ; C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO'
3677 ;      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
3678 ;      THAT RMER1 HAS NOT BEEN CLEARED.
3679 ;
3680 ; D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
3681 ;
3682 ; E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
3683 ;      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3684 ;
3685 ;*****

```

```

3686 015622          TST12:
3687 015622 005737 001300      TST    KYBCTL        ; PERFORMING ONLY SINGLE TESTS ?
3688 015626 001406              BEQ    2$            ; BR IF NOT
3689 015630 100002              BPL    1$            ; BR IF JUST ENTERED TEST
3690 015632 000137 002676      JMP    EXEC          ; RETURN & GET NEXT TEST NUMBER
3691 015636 012737 177777 001300 1$:  MOV    #-1, KYBCTL   ; SET SINGLE TEST INDICATOR
3692 015644 112737 000012 001102 2$:  MOVB   #12, $STNM    ; TEST NUMBER
3693 015652 012737 015674 001106      MOV    #TEST12, $LPADR ; LOAD LOOP ON TEST ADDRESS

```



```

3694 015660 012737 015674 001110      MOV      #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3695 015666 012737 000031 001176      MOV      #25,$TIMES ;DO 25. ITERATIONS
3696 015674 012706 001100      TEST12: MOV      #STACK,$P ;LOAD THE STACK POINTER
3697
3698 ;CLEAR ATTENTION BITS FOR BOTH PORTS
3699
3700 015700 113760 001224 000010      MOVB     PORTA,$MCS2($R0) ;SELECT PORT #A
3701 015706 005060 000012      CLR      $RMS1($R0) ;SEIZE THE DRIVE
3702 015712 012760 000011 000000      MOV      #11,$MCS1($R0) ;ISSUE DRIVE CLEAR
3703 015720 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE
3704 015726 113760 001226 000010      MOVB     PORTB,$MCS2($R0) ;SELECT PORT #B
3705 015734 005060 000012      CLR      $RMS1($R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
3706 015740 012760 000011 000000      MOV      #11,$MCS1($R0) ;ISSUE DRIVE CLEAR
3707 015746 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE
3708 ;*****
3709
3710 ;SEIZE THE DRIVE THROUGH PORT A
3711
3712 015754 113760 001224 000010      MOVB     PORTA,$MCS2($R0) ;SELECT PORT A
3713 015762 013737 001224 001242      MOV      PORTA,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
3714 015770 005060 000012      CLR      $RMS1($R0) ;WRITE RMS1
3715 015774 013737 001226 001244      MOV      PORTB,$OPPT ;'OPPOSITE' PORT ADDRESS
3716
3717 ;*****
3718 ;FORCE AN ERROR
3719
3720 016002 012760 177777 000014      MOV      #-1,$RMR1($R0) ;SET ERROR BITS
3721 016010 012760 000013 000000      MOV      #13,$MCS1($R0) ;ISSUE A RELEASE COMMAND
3722 016016 005037 001250      CLR      $CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3723 016022 016037 000000 001126      MOV      $RMS1($R0),$SBOADR ;GET CONTENTS OF RMS1
3724 016030 012737 000000 001122      MOV      #RMS1,$SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3725 016036 060037 001122      ADD      $R0,$SBOADR ;ADD RH11 BASE ADDRESS
3726 016042 012737 004012 001124      MOV      #4012,$SGDAT ;WHAT REGISTER SHOULD BE
3727 016050 013737 001126 001164      MOV      $SBOADR,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
3728 016056 042737 173765 001164      BIC      #4012,$STMP0 ;SAVE SPECIFIED BITS
3729 016064 023737 001124 001164      CMP      $SGDAT,$STMP0 ;COMPARE THE BITS
3730 016072 001414      BEQ      $66$ ;BR IF OK
3731 016074 013737 001126 001174      MOV      $SBOADR,$STMP4 ;COPY 'BAD DATA'
3732 016102 042737 004012 001174      BIC      #4012,$STMP4 ;CLEAR THE MASKED BITS
3733 016110 053737 001174 001124      BIS      $STMP4,$SGDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3734 016116 104025      ERROR   25 ;TYPE MESSAGE 25
3735 016120 005137 001250      COM      $CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3736 016124 000240      66$:   NOP
3737 016126 005737 001250      TST      $CKERR ;DID 'GO' BIT RESET ?
3738 016132 001002      BNE      .+6 ;BR IF NOT
3739 016134 000137 016174      JMP      1$ ;'GO' BIT RESET
3740 016140 012760 000040 000010      MOV      #CLR,$MCS2($R0) ;INIT THE RH11
3741 016146 113760 001224 000010      MOVB     PORTA,$MCS2($R0) ;SELECT PORT A
3742 016154 013737 001224 001240      MOV      PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3743 016162 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE THROUGH PORT A
3744 016170 000137 016740      JMP      2$ ;BYPASS THE REST OF THE TEST
3745
3746 ;*****
3747 ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
3748
3749 016174      1$:

```

```

3750 016174 :13760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3751 016202 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3752 016210 005037 001250      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3753 016214 016037 000012 001126      MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
3754 016222 012737 000012 001122      MOV    #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3755 016230 060037 001122      ADD    RO, $B0ADR ;ADD RH11 BASE ADDRESS
3756 016234 005037 001124      CLR    $GDDAT ;WHAT REGISTER SHOULD BE
3757 016240 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
3758 016246 001403      BEQ    68$ ;BR IF OK
3759 016250 104024      ERROR  24 ;TYPE MESSAGE 24
3760 016252 005137 001250      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3761 016256 000240      NOP
3762 016260 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3763 016266 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3764 016274 005037 001250      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3765 016300 016037 000014 001126      MOV    RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
3766 016306 012737 000014 001122      MOV    #RMER1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3767 016314 060037 001122      ADD    RO, $B0ADR ;ADD RH11 BASE ADDRESS
3768 016320 012737 177777 001124      MOV    #177777, $GDDAT ;WHAT REGISTER SHOULD BE
3769 016326 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
3770 016334 001403      BEQ    70$ ;BR IF OK
3771 016336 104010      ERROR  10 ;REPORT THE ERROR
3772 016340 005137 001250      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3773 016344 000240      NOP
3774
3775 ;*****
3776 ;CLEAR THE ERRORS THROUGH PORT A
3777
3778 016346 012760 000011 000000      MOV    #11, RMCS1(RO) ;ISSUE A DRIVE CLEAR
3779
3780 ;*****
3781
3782 ;RELEASE THE DRIVE FROM PORT A
3783
3784 016354 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3785 016362 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3786 016370 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
3787
3788 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3789
3790 016376 005037 001254      CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3791 016402 012737 000012 001122      MOV    #RMDS1, $B0ADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3792 016410 060037 001122      ADD    RO, $B0ADR ;ADD THE I/O BASE ADDRESS
3793 016414 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3794 016422 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
3795 016430 016037 000012 001170      MOV    RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3796 016436 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
3797 016444 013737 001170 001164      MOV    $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
3798 016452 042737 100100 001164      BIC    #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3799 016460 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
3800 016466 016037 000012 001172      MOV    RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3801 016474 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
3802 016502 013737 001172 001166      MOV    $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
3803 016510 042737 100100 001166      BIC    #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3804 016516 023737 001164 001166      CMP    $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3805 016524 001006      BNE    72$ ;BR IF NOT

```

```

3806 016526 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3807 016532 001045      BNE      74$        ;BR IF NOT
3808 016534 104046      ERROR   46         ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3809 016536 000137 016736      JMP      76$        ;BYPASS THE REST OF THE CHECKS
3810 016542 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3811 016550 013737 001226 001240      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3812 016556 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B.
3813 016564 005737 001164      TST     $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
3814 016570 001414      BEQ     73$        ;BR IF ZERO
3815 016572 013737 001224 001240      MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3816 016600 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3817 016606 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.
3818 016614 005737 001166      TST     $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
3819 016620 001012      BNE     74$        ;BR IF NOT
3820 016622 012737 177777 001254 73$:  MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3821 016630 012760 000011 000000      MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
3822 016636 012760 000013 000000      MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
3823 016644 104026      ERROR   26         ;TYPE ERROR MESSAGE 26
3824 016646 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD51 READ
3825 016654 013737 001224 001240      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
3826 016662 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3827 016670 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
3828 016676 001401      BEQ     75$        ;BR IF OK FROM PORT A.
3829 016700 104007      ERROR   7          ;REPORT ERROR
3830 016702 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT ;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
3831 016710 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
3832 016716 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3833 016724 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
3834 016732 001401      BEQ     76$        ;BR IF OK
3835 016734 104007      ERROR   7          ;REPORT ERROR
3836 016736 000240 76$:  NOP
3837 016740 000004 2$:  SCOPE ;LOOP ?

```

```

3838
3839
3840 ;*****
3841 ;TEST 13      PORT 'B' RELEASE W/ERRORS TEST
3842 ;
3843 ;VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
3844 ;BITS ARE SET IN THE DRIVE.
3845 ;
3846 ; A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
3847 ;
3848 ; B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
3849 ;
3850 ; C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO'
3851 ; BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
3852 ; THAT RMER1 HAS NOT BEEN CLEARED.
3853 ;
3854 ; D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
3855 ;
3856 ; E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
3857 ; RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3858 ;*****

```

```

3859 016742 005737 001300 †ST13:
3860 016742 001406      TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
3861 016746      BEQ     2$         ;BR IF NOT

```

```

3862 016750 100002          BPL      1$          ;BR IF JUST ENTERED TEST
3863 016752 000137 002676    JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
3864 016756 012737 177777 001300 1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3865 016764 112737 000013 001102 2$:      MOVVB   #13,$STSTNM ;TEST NUMBER
3866 016772 012737 017014 001106      MOV      #TEST13,$LPAOR ;LOAD LOOP ON TEST ADDRESS
3867 017000 012737 017014 001110      MOV      #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3868 017006 012737 000031 001176      MOV      #25,$TIMES    ;DO 25. ITERATIONS
3869 017014 012706 001100  TEST13:  MOV      #STACK,$P    ;LOAD THE STACK POINTER
3870
3871          ;CLEAR ATTENTION BITS FOR BOTH PORTS
3872
3873 017020 113760 001224 000010  MOVVB   PORTA,$MCS2($R0) ;SELECT PORT #A
3874 017026 005060 000012      CLR      $RMS1($R0)    ;SEIZE THE DRIVE
3875 017032 012760 000011 000000      MOV      #11,$MCS1($R0) ;ISSUE DRIVE CLEAR
3876 017040 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE
3877 017046 113760 001226 000010  MOVVB   PORTB,$MCS2($R0) ;SELECT PORT #B
3878 017054 005060 000012      CLR      $RMS1($R0)    ;SEIZE THE DRIVE THROUGH PORT 'B'
3879 017060 012760 000011 000000      MOV      #11,$MCS1($R0) ;ISSUE DRIVE CLEAR
3880 017066 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE
3881          ;*****
3882
3883          ;SEIZE THE DRIVE THROUGH PORT B
3884
3885 017074 113760 001226 000010  MOVVB   PORTB,$MCS2($R0) ;SELECT PORT B
3886 017102 013737 001226 001242      MOV      PORTB,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
3887 017110 005060 000012      CLR      $RMS1($R0)    ;WRITE RMS1
3888 017114 013737 001224 001244      MOV      PORTA,$OPPRT  ;'OPPOSITE' PORT ADDRESS
3889
3890          ;*****
3891          ;FORCE AN ERROR
3892
3893 017122 012760 177777 000014      MOV      #-1,$MER1($R0) ;SET ERROR BITS
3894 017130 012760 000013 000000      MOV      #13,$MCS1($R0) ;ISSUE A RELEASE COMMAND
3895 017136 005037 001250      CLR      $CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
3896 017142 016037 000000 001126      MOV      $RMS1($R0),$SBDADR ;GET CONTENTS OF RMS1
3897 017150 012737 000000 001122      MOV      #RMS1,$SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3898 017156 060037 001122      ADD      $R0,$SBDADR   ;ADD RH11 BASE ADDRESS
3899 017162 012737 004012 001124      MOV      #4012,$GDDAT  ;WHAT REGISTER SHOULD BE
3900 017170 013737 001126 001164      MOV      $SBDADR,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
3901 017176 042737 173765 001164      BIC      #1C4012,$STMP0 ;SAVE SPECIFIED BITS
3902 017204 023737 001124 001164      CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
3903 017212 001414      BEQ      66$          ;BR IF OK
3904 017214 013737 001126 001174      MOV      $SBDADR,$STMP4 ;COPY 'BAD DATA'
3905 017222 042737 004012 001174      BIC      #4012,$STMP4  ;CLEAR THE MASKED BITS
3906 017230 053737 001174 001124      BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3907 017236 104025      ERROR   25          ;TYPE MESSAGE 25
3908 017240 005137 001250      COM      $CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
3909 017244 000240          66$:      NOP
3910 017246 005737 001250      TST      $CKERR       ;DID 'GO' BIT RESET ?
3911 017252 001002      BNE      .+6         ;BR IF NOT
3912 017254 000137 017314      JMP      1$          ;'GO' BIT RESET
3913 017260 012760 000040 000010      MOV      #CLR,$MCS2($R0) ;INIT THE RH11
3914 017266 113760 001226 000010      MOVVB   PORTB,$MCS2($R0) ;SELECT PORT B
3915 017274 013737 001226 001240      MOV      PORTB,$PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3916 017302 012760 000013 000000      MOV      #13,$MCS1($R0) ;RELEASE THE DRIVE THROUGH PORT B
3917 017310 000137 020060      JMP      2$          ;BYPASS THE REST OF THE TEST
    
```

K06

```

3918
3919 ;*****
3920 ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
3921
3922 017314 113760 001224 000010 15:      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3923 017314 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3924 017322 005037 001250      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3925 017330 016037 000012 001126      MOV   RMDS1(RO), $BODAT ;GET CONTENTS OF RMDS1
3926 017334 012737 000012 001122      MOV   #RMDS1, $BODADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3927 017342 005037 001124 001126      ADD   RO, $BODADR ;ADD RH11 BASE ADDRESS
3928 017350 023737 001124 001126      CLR   $GDDAT ;WHAT REGISTER SHOULD BE
3929 017354 005037 001124 001126      CMP   $GDDAT, $BODAT ;IS THE REGISTER OK ?
3930 017360 001403 001124 001126      BEQ   68$ ;BR IF OK
3931 017366 104024      ERROR 24 ;TYPE MESSAGE 24
3932 017370 005137 001250      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3933 017372 000240      68$:  NOP
3934 017376 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3935 017400 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3936 017406 005037 001250      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3937 017414 016037 000014 001126      MOV   RMER1(RO), $BODAT ;GET CONTENTS OF RMER1
3938 017420 012737 000014 001122      MOV   #RMER1, $BODADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3939 017426 060037 001122      ADD   RO, $BODADR ;ADD RH11 BASE ADDRESS
3940 017434 012737 177777 001124      MOV   #177777, $GDDAT ;WHAT REGISTER SHOULD BE
3941 017440 023737 001124 001126      CMP   $GDDAT, $BODAT ;IS THE REGISTER OK ?
3942 017446 001403 001124 001126      BEQ   70$ ;BR IF OK
3943 017454 104010      ERROR 10 ;REPORT THE ERROR
3944 017456 005137 001250      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3945 017460 000240      70$:  NOP
3946 017464
3947
3948 ;*****
3949 ;CLEAR THE ERRORS THROUGH PORT B
3950
3951 017466 012760 000011 000000      MOV   #11, RMCS1(RO) ;ISSUE A DRIVE CLEAR
3952
3953 ;*****
3954 ;RELEASE THE DRIVE FROM PORT B
3955
3956
3957 017474 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3958 017502 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3959 017510 012760 000013 000000      MOV   #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3960
3961 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3962
3963 017516 005037 001254      CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3964 017522 012737 000012 001122      MOV   #RMDS1, $BODADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3965 017530 060037 001122      ADD   RO, $BODADR ;ADD THE I/O BASE ADDRESS
3966 017534 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3967 017542 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
3968 017550 016037 000012 001170      MOV   RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3969 017556 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
3970 017564 013737 001170 001164      MOV   $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
3971 017572 042737 100100 001164      BIC   #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3972 017600 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
3973 017606 016037 000012 001172      MOV   RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

```

3974 017614 042737 024001 001172 BIC #PIP!WRL!OM,STMP3 ;CLEAR CNT CARES
3975 017622 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
3976 017630 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3977 017636 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3978 017644 001006 BNE 72$ ;BR IF NOT
3979 017646 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3980 017652 001045 BNE 74$ ;BR IF NOT
3981 017654 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3982 017656 000137 020056 JMP 76$ ;BYPASS THE REST OF THE CHECKS
3983 017662 013737 001170 001126 72$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3984 017670 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3985 017676 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
3986 017704 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3987 017710 001414 BEQ 73$ ;BR IF ZERO
3988 017712 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3989 017720 013737 001172 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3990 017726 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
3991 017734 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3992 017740 001012 BNE 74$ ;BR IF NOT
3993 017742 012737 177777 001254 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3994 017750 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
3995 017756 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
3996 017764 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
3997 017766 013737 001170 001126 74$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3998 017774 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
3999 020002 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4000 020010 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
4001 020016 001401 BEQ 75$ ;BR IF OK FROM PORT A.
4002 020020 104007 ERROR 7 ;REPORT ERROR
4003 020022 013737 001172 001126 75$: MOV STMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4004 020030 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4005 020036 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4006 020044 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4007 020052 001401 BEQ 76$ ;BR IF OK
4008 020054 104007 ERROR 7 ;REPORT ERROR
4009 020056 000240 76$: NOP
4010 020060 000004 2$: SCOPE ;LOOP ?

```

```

4011
4012
4013 *****
4014 *TEST 14 PORT 'A' SEIZE AND CLEAR TEST
4015 *
4016 *VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
4017 * PORT TO RELEASE THE DRIVE.
4018 *
4019 * A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS1 THROUGH PORT 'A'.
4020 * VERIFY THAT THE DRIVE HAS BEEN SEIZED.
4021 *
4022 * B. ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
4023 * DOES NOT RETURN TO NEUTRAL.
4024 *
4025 * C. ISSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE
4026 * DOES NOT RETURN TO NEUTRAL.
4027 *
4028 * D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
4029 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

```

```

4030 ;*
4031 ;:*****
4032 020062 TST14: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
4033 020062 005737 001300 BEQ 25 ;BR IF NOT
4034 020066 001406 BPL 15 ;BR IF JUST ENTERED TEST
4035 020070 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
4036 020072 000137 002676 15: MOV #1, KYBCTL ;SET SINGLE TEST INDICATOR
4037 020076 012737 177777 001300 25: MOV #14, STSTNM ;TEST NUMBER
4038 020104 112737 000014 001102 MOV #TEST14, SLPADR ;LOAD LOOP ON TEST ADDRESS
4039 020112 012737 020134 001106 MOV #TEST14, SLPERR ;LOAD LOOP ON ERROR ADDRESS
4040 020120 012737 020134 001110 MOV #25, STTIMES ;DO 25. ITERATIONS
4041 020126 012737 000031 001176 TEST14: MOV #STACK, SP ;LOAD THE STACK POINTER
4042 020134 012706 001100
4043
4044 ;:*****
4045
4046 ;SEIZE THE DRIVE THROUGH PORT A
4047
4048 020140 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4049 020146 013737 001224 001242 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
4050 020154 005060 000012 CLR RMDS1(RO) ;WRITE RMDS1
4051 020160 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4052 020166 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4053 020174 013737 001226 001244 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
4054 020202 016037 000012 001126 MOV RMDS1(RO), SBDAT ;SEE IF DRIVE SEIZED BY PORT A
4055 020210 010037 001122 MOV RO, SBDADR ;R11 BASE ADDRESS
4056 020214 062737 000012 001122 ADD #RMDS1, SBDADR ;GENERATE BAD REGISTER ADDRESS
4057 020222 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
4058 020226 023737 001124 001126 CMP $GDDAT, SBDAT ;IS THE REGISTER ZERO
4059 020234 001403 BEQ 645 ;BR IF IT IS
4060 020236 104004 ERROR 4 ;REPORT THE ERROR
4061 020240 000137 021460 JMP 15 ;BYPASS REST OF THE SUBTEST
4062 020244
4063 020244 113760 001224 000010 645: MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4064 020252 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4065 020260 016037 000012 001126 MOV RMDS1(RO), SBDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4066 020266 042737 020001 001126 BIC #OM!PIP, SBDAT ;CLEAR DONT CARE BITS
4067 020274 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
4068 020302 013737 001124 001166 MOV $GDDAT, STMP1 ;USE GOOD DATA AS A MASK
4069 020310 005137 001166 COM STMP1 ;COMPLEMENT THE EXPECTED STATUS
4070 020314 013737 001126 001164 MOV SBDAT, STMPO ;SAVE THE ACTUAL STATUS
4071 020322 043737 001166 001164 BIC STMP1, STMPO ;CLEAR UNWANTED BITS
4072 020330 023737 001124 001164 CMP $GDDAT, STMPO ;ARE THE EXPECTED STATUS BITS SET ?
4073 020336 001401 BEQ 655 ;BR IF THEY ARE
4074 020340 104005 ERROR 5 ;REPORT THE ERROR
4075 020342 000240 655: NOP
4076
4077 ;:*****
4078 ;DRIVE CLEAR THROUGH PORT A FIRST
4079
4080 020344 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT A
4081
4082 ;:*****
4083 ;VERIFY THAT DRIVE STILL SEIZED BY PORT A
4084
4085 020352 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B

```

```

4086 020360 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4087 020366 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4088 020372 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
4089 020400 012737 000012 001122 MOV #RMD51,SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4090 020406 060037 001122 ADD RO,SBOADR ;ADD RH11 BASE ADDRESS
4091 020412 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
4092 020416 013737 001126 001164 MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4093 020424 042737 100000 001164 BIC #1C7777,STMP0 ;SAVE SPECIFIED BITS
4094 020432 023737 001124 001164 CMP $GDDAT,STMP0 ;COMPARE THE BITS
4095 020440 001414 BEQ 66$ ;BR IF OK
4096 020442 013737 001126 001174 MOV SBDAT,STMP4 ;COPY 'BAD DATA'
4097 020450 042737 077777 001174 BIC #7777,STMP4 ;CLEAR THE MASKED BITS
4098 020456 053737 001174 001124 BIS STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4099 020464 104033 ERROR 33 ;TYPE MESSAGE 33
4100 020466 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4101 020472 000240 66$: NOP
4102 020474 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
4103 020502 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4104 020510 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4105 020514 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
4106 020522 012737 000012 001122 MOV #RMD51,SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4107 020530 060037 001122 ADD RO,SBOADR ;ADD RH11 BASE ADDRESS
4108 020534 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
4109 020542 013737 001126 001164 MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4110 020550 042737 100000 001164 BIC #1C7777,STMP0 ;SAVE SPECIFIED BITS
4111 020556 023737 001124 001164 CMP $GDDAT,STMP0 ;COMPARE THE BITS
4112 020564 001414 BEQ 68$ ;BR IF OK
4113 020566 013737 001126 001174 MOV SBDAT,STMP4 ;COPY 'BAD DATA'
4114 020574 042737 077777 001174 BIC #7777,STMP4 ;CLEAR THE MASKED BITS
4115 020602 053737 001174 001124 BIS STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4116 020610 104033 ERROR 33 ;TYPE MESSAGE 33
4117 020612 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4118 020616 000240 68$: NOP
4119
4120 ;:*****
4121 ;NOW ISSUE MASSBUS INIT
4122
4123 020620 012760 000040 000010 MOV #CLR,RMCS2(RO) ;ISSUE MASSBUS INIT
4124
4125 ;:*****
4126 ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
4127
4128 020626 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
4129 020634 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4130 020642 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4131 020646 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
4132 020654 012737 000012 001122 MOV #RMD51,SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4133 020662 060037 001122 ADD RO,SBOADR ;ADD RH11 BASE ADDRESS
4134 020666 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
4135 020672 013737 001126 001164 MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4136 020700 042737 100000 001164 BIC #1C7777,STMP0 ;SAVE SPECIFIED BITS
4137 020706 023737 001124 001164 CMP $GDDAT,STMP0 ;COMPARE THE BITS
4138 020714 001414 BEQ 70$ ;BR IF OK
4139 020716 013737 001126 001174 MOV SBDAT,STMP4 ;COPY 'BAD DATA'
4140 020724 042737 077777 001174 BIC #7777,STMP4 ;CLEAR THE MASKED BITS
4141 020732 053737 001174 001124 BIS STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```



```

4142 020740 104034          ERROR 34          ;TYPE MESSAGE 34
4143 020742 005137 001250    COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
4144 020746 000240          NOP
4145 020750 113760 001224 000010 705:  MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
4146 020756 013737 001224 001240    MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4147 020764 005037 001250          CLR  CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
4148 020770 016037 000012 001126    MOV  RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
4149 020776 012737 000012 001122    MOV  #RMD51, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4150 021004 060037 001122          ADD  RO, $BDDADR ;ADD R/11 BASE ADDRESS
4151 021010 012737 011700 001124    MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
4152 021016 013737 001126 001164    MOV  $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
4153 021024 042737 100000 001164    BIC  #1C77777, $TMP0 ;SAVE SPECIFIED BITS
4154 021032 023737 001124 001164    CMP  $GDDAT, $TMP0 ;COMPARE THE BITS
4155 021040 001414          BEQ  725
4156 021042 013737 001126 001174    MOV  $BDDAT, $TMP4 ;COPY 'BAD DATA'
4157 021050 042737 077777 001174    BIC  #77777, $TMP4 ;CLEAR THE MASKED BITS
4158 021056 053737 001174 001124    BIS  $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4159 021064 104034          ERROR 34          ;TYPE MESSAGE 34
4160 021066 005137 001250    COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
4161 021072 000240          NOP
4162
4163          ;RELEASE THE DRIVE FROM PORT A
4164
4165 021074 113760 001224 000010  MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
4166 021102 013737 001224 001240    MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4167 021110 012760 000013 000000    MOV  #13, RMD51(RO) ;ISSUE RELEASE THROUGH PORT A
4168
4169          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4170
4171 021116 005037 001254          CLR  RELERR     ;CLEAR THE 'RELEASE ERROR' INDICATOR
4172 021122 012737 000012 001122    MOV  #RMD51, $BDDADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
4173 021130 060037 001122          ADD  RO, $BDDADR ;ADD THE I/O BASE ADDRESS
4174 021134 012737 011700 001124    MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
4175 021142 113760 001224 000010  MOVB  PORTA, RMCS2(RO) ;SELECT PORT A.
4176 021150 016037 000012 001170    MOV  RMD51(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4177 021156 042737 024001 001170    BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
4178 021164 013737 001170 001164    MOV  $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
4179 021172 042737 100100 001164    BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4180 021200 113760 001226 000010  MOVB  PORTB, RMCS2(RO) ;SELECT PORT B.
4181 021206 016037 000012 001172    MOV  RMD51(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4182 021214 042737 024001 001172    BIC  #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
4183 021222 013737 001172 001166    MOV  $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
4184 021230 042737 100100 001166    BIC  #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4185 021236 023737 001164 001166    CMP  $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4186 021244 001006          BNE  745
4187 021246 005737 001164          TST  $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4188 021252 001045          BNE  765
4189 021254 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4190 021256 000137 021456          JMP  785
4191 021262 013737 001170 001126 745:  MOV  $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4192 021270 013737 001226 001240    MOV  PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4193 021276 113760 001226 000010  MOVB  PORTB, RMCS2(RO) ;SELECT PORT B.
4194 021304 005737 001164          TST  $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4195 021310 001414          BEQ  755
4196 021312 013737 001224 001240    MOV  PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4197 021320 013737 001172 001126    MOV  $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
    
```

```

4198 021326 113760 001224 000010      MOVB  PORTA, RMCS2(RD) ; SELECT PORT A.
4199 021334 005737 001166              TST   $TMP1           ; SEE IF STATUS EQ ZERO FROM PORT B.
4200 021340 001012              BNE   76$            ; BR IF NOT
4201 021342 012737 177777 001254 75$:  MOV   #-1, RELERR    ; SET 'RELEASE ERROR' INDICATOR
4202 021350 012760 000011 000000      MOV   #11, RMCS1(RD) ; CLEAR THE DRIVE
4203 021356 012760 000013 000000      MOV   #13, RMCS1(RD) ; RELEASE THE DRIVE
4204 021364 104026              ERROR 26             ; TYPE ERROR MESSAGE 26
4205 021366 013737 001170 001126 76$:  MOV   $TMP2, $BDDAT  ; LOOK FOR BIT FAILURES WHEN RMDS1 READ
4206 021374 013737 001224 001240      MOV   PORTA, PTNBR   ; CHANGE PORT NUMBER
4207 021402 042737 100000 001126      BIC   #ATA, $BDDAT   ; DON'T CHECK THE ATTN BIT
4208 021410 023737 001124 001126      CMP   $GDDAT, $BDDAT ; ALL BITS OK ?
4209 021416 001401              BEQ   77$            ; BR IF OK FROM PORT A.
4210 021420 104007              ERROR 7              ; REPORT ERROR
4211 021422 013737 001172 001126 77$:  MOV   $TMP3, $BDDAT  ; CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4212 021430 013737 001226 001240      MOV   PORTB, PTNBR   ; CHANGE PORT NUMBER
4213 021436 042737 100000 001126      BIC   #ATA, $BDDAT   ; DON'T CHECK THE ATTN BIT
4214 021444 023737 001124 001126      CMP   $GDDAT, $BDDAT ; SEE IF READ OK FROM PORT B.
4215 021452 001401              BEQ   78$            ; BR IF OK
4216 021454 104007              ERROR 7              ; REPORT ERROR
4217 021456 000240 78$:  NOP
4218 021460 000004 1$:  SCOPE                ; LOOP ?

```

```

4219
4220 ;*****
4221 ;*TEST 15      PORT 'B' SEIZE AND CLEAR TEST
4222 ;*
4223 ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
4224 ;*      PORT TO RELEASE THE DRIVE.
4225 ;*
4226 ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RMDS1 THROUGH PORT 'B'.
4227 ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
4228 ;*
4229 ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
4230 ;*      DOES NOT RETURN TO NEUTRAL.
4231 ;*
4232 ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE
4233 ;*      DOES NOT RETURN TO NEUTRAL.
4234 ;*
4235 ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
4236 ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4237 ;*
4238 ;*****

```

```

4239 021462
4240 021462 005737 001300      TST   KYBCTL         ; PERFORMING ONLY SINGLE TESTS ?
4241 021466 001406              BEQ   2$             ; BR IF NOT
4242 021470 100002              BPL   1$             ; BR IF JUST ENTERED TEST
4243 021472 000137 002676      JMP   EXEC           ; RETURN & GET NEXT TEST NUMBER
4244 021476 012737 177777 001300 1$:  MOV   #-1, KYBCTL    ; SET SINGLE TEST INDICATOR
4245 021504 112737 000015 001102 2$:  MOVB  #15, $STNM     ; TEST NUMBER
4246 021512 012737 021534 001106      MOV   #TEST15, $LPADR ; LOAD LOOP ON TEST ADDRESS
4247 021520 012737 021534 001110      MOV   #TEST15, $LPERR ; LOAD LOOP ON ERROR ADDRESS
4248 021526 012737 000031 001176      MOV   #25, $TIMES    ; DO 25. ITERATIONS
4249 021534 012706 001100 TEST15: MOV  #STACK, SP    ; LOAD THE STACK POINTER

```

```

4250
4251 ;*****
4252 ;
4253 ;SEIZE THE DRIVE THROUGH PORT B

```

```

4254
4255 021540 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
4256 021546 013737 001226 001242      MOV   PORTB, SEIZPT ; STORE SEIZING PORT'S ADDRESS
4257 021554 005060 000012      CLR   RMD51(RO) ; WRITE RMD51
4258 021560 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
4259 021566 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4260 021574 013737 001224 001244      MOV   PORTA, OPPRT ; 'OPPOSITE' PORT ADDRESS
4261 021602 016037 000012 001126      MOV   RMD51(RO), $BDDAT ; SEE IF DRIVE SEIZED BY PORT B
4262 021610 010037 001122      MOV   RO, $BOADR ; RH11 BASE ADDRESS
4263 021614 062737 000012 001122      ADD   #RMD51, $BOADR ; GENERATE BAD REGISTER ADDRESS
4264 021622 005037 001124      CLR   $GDDAT ; REGISTER SHOULD BE ZERO
4265 021626 023737 001124 001126      CMP   $GDDAT, $BDDAT ; IS THE REGISTER ZERO
4266 021634 001403      BEQ   64$ ; BR IF IT IS
4267 021636 104004      ERROR 4 ; REPORT THE ERROR
4268 021640 000137 023060      JMP   1$ ; BYPASS REST OF THE SUBTEST
4269 021644
4270 021644 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
4271 021652 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4272 021660 016037 000012 001126      MOV   RMD51(RO), $BDDAT ; SEE IF SEIZING PORT SEES CORRECT STATUS
4273 021666 042737 020001 001126      BIC   #OM!PIP, $BDDAT ; CLEAR DONT CARE BITS
4274 021674 012737 011700 001124      MOV   #MOL!PGH!DPR!DRY!VV, $GDDAT ; EXPECTED STATUS
4275 021702 013737 001124 001166      MOV   $GDDAT, $TMP1 ; USE GOOD DATA AS A MASK
4276 021710 005137 001166      COM   $TMP1 ; COMPLEMENT THE EXPECTED STATUS
4277 021714 013737 001126 001164      MOV   $BDDAT, $TMP0 ; SAVE THE ACTUAL STATUS
4278 021722 043737 001166 001164      BIC   $TMP1, $TMP0 ; CLEAR UNWANTED BITS
4279 021730 023737 001124 001164      CMP   $GDDAT, $TMP0 ; ARE THE EXPECTED STATUS BITS SET ?
4280 021736 001401      BEQ   65$ ; BR IF THEY ARE
4281 021740 104005      ERROR 5 ; REPORT THE ERROR
4282 021742 000240
4283
4284
4285 ;*****
4286 ;DRIVE CLEAR THROUGH PORT B FIRST
4287 021744 012760 000011 000000      MOV   #11, RMCS1(RO) ; ISSUE DRIVE CLEAR THROUGH PORT B
4288
4289 ;*****
4290 ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
4291
4292 021752 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
4293 021760 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4294 021766 005037 001250      CLR   CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
4295 021772 016037 000012 001126      MOV   RMD51(RO), $BDDAT ; GET CONTENTS OF RMD51
4296 022000 012737 000012 001122      MOV   #RMD51, $BOADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
4297 022006 060037 001122      ADD   RO, $BOADR ; ADD RH11 BASE ADDRESS
4298 022012 005037 001124      CLR   $GDDAT ; WHAT REGISTER SHOULD BE
4299 022016 013737 001126 001164      MOV   $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO 'TMP0'
4300 022024 042737 100000 001164      BIC   #1C7777, $TMP0 ; SAVE SPECIFIED BITS
4301 022032 023737 001124 001164      CMP   $GDDAT, $TMP0 ; COMPARE THE BITS
4302 022040 001414      BEQ   66$ ; BR IF OK
4303 022042 013737 001126 001174      MOV   $BDDAT, $TMP4 ; COPY 'BAD DATA'
4304 022050 042737 077777 001174      BIC   #77777, $TMP4 ; CLEAR THE MASKED BITS
4305 022056 053737 001174 001124      BIS   $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
4306 022064 104033      ERROR 33 ; TYPE MESSAGE 33
4307 022066 005137 001250      COM   CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
4308 022072 000240
4309 022074 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B

```

64\$:

65\$:

66\$:

```

4310 022102 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4311 022110 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
4312 022114 016037 000012 001126      MOV      RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
4313 022122 012737 000012 001122      MOV      #RMD51,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4314 022130 060037 001122      ADD      RO,SBADR   ;ADD RH11 BASE ADDRESS
4315 022134 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
4316 022142 013737 001126 001164      MOV      SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4317 022150 042737 100000 001164      BIC      #1C7777,$TMP0 ;SAVE SPECIFIED BITS
4318 022156 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
4319 022164 001414      BEQ      68$        ;BR IF OK
4320 022166 013737 001126 001174      MOV      SBDDAT,$TMP4 ;COPY 'BAD DATA'
4321 022174 042737 077777 001174      BIC      #77777,$TMP4 ;CLEAR THE MASKED BITS
4322 022202 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4323 022210 104033      ERROR   33        ;TYPE MESSAGE 33
4324 022212 005137 001250      COM      CKERR     ;SET THE REGISTER COMPARE ERROR INDICATOR
4325 022216 000240      NOP
4326
4327
4328
4329
4330 022220 012760 000040 000010      MOV      #CLR,RMCS2(RO) ;ISSUE MASSBUS INIT
4331
4332
4333
4334
4335 022226 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
4336 022234 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4337 022242 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
4338 022246 016037 000012 001126      MOV      RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
4339 022254 012737 000012 001122      MOV      #RMD51,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4340 022262 060037 001122      ADD      RO,SBADR   ;ADD RH11 BASE ADDRESS
4341 022266 005037 001124      CLR      $GDDAT     ;WHAT REGISTER SHOULD BE
4342 022272 013737 001126 001164      MOV      SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4343 022300 042737 100000 001164      BIC      #1C7777,$TMP0 ;SAVE SPECIFIED BITS
4344 022306 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
4345 022314 001414      BEQ      70$        ;BR IF OK
4346 022316 013737 001126 001174      MOV      SBDDAT,$TMP4 ;COPY 'BAD DATA'
4347 022324 042737 077777 001174      BIC      #77777,$TMP4 ;CLEAR THE MASKED BITS
4348 022332 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4349 022340 104034      ERROR   34        ;TYPE MESSAGE 34
4350 022342 005137 001250      COM      CKERR     ;SET THE REGISTER COMPARE ERROR INDICATOR
4351 022346 000240      NOP
4352 022350 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
4353 022356 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4354 022364 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
4355 022370 016037 000012 001126      MOV      RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
4356 022376 012737 000012 001122      MOV      #RMD51,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4357 022404 060037 001122      ADD      RO,SBADR   ;ADD RH11 BASE ADDRESS
4358 022410 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
4359 022416 013737 001126 001164      MOV      SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4360 022424 042737 100000 001164      BIC      #1C7777,$TMP0 ;SAVE SPECIFIED BITS
4361 022432 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
4362 022440 001414      BEQ      72$        ;BR IF OK
4363 022442 013737 001126 001174      MOV      SBDDAT,$TMP4 ;COPY 'BAD DATA'
4364 022450 042737 077777 001174      BIC      #77777,$TMP4 ;CLEAR THE MASKED BITS
4365 022456 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

68\$:

70\$:

72\$:

4366	022464	104034				ERROR	34		;TYPE MESSAGE 34
4367	022466	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
4368	022472	000240			72\$:	NOP			
4369									
4370									;RELEASE THE DRIVE FROM PORT B
4371									
4372	022474	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B
4373	022502	013737	001226	001240		MOV	PORTB, PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4374	022510	012760	000013	000000		MOV	#13, RMCS1(RO)		;ISSUE RELEASE THROUGH PORT B
4375									
4376									;VERIFY THAT THE DRIVE IS IN NEUTRAL
4377									
4378	022516	005037	001254			CLR	RELERR		;CLEAR THE 'RELEASE ERROR' INDICATOR
4379	022522	012737	000012	001122		MOV	#RMD51, \$BDDAT		;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
4380	022530	060037	001122			ADD	RO, \$BDDAT		;ADD THE I/O BASE ADDRESS
4381	022534	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT		;COMPARISON CONSTANT
4382	022542	113760	001224	000010		MOVB	PORTA, RMCS2(RO)		;SELECT PORT A.
4383	022550	016037	000012	001170		MOV	RMD51(RO), \$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.
4384	022556	042737	024001	001170		BIC	#PIP!WRL!OM, \$TMP2		;CLEAR DONT CARES
4385	022564	013737	001170	001164		MOV	\$TMP2, \$TMP0		;COPY IT INTO 'TMP0'
4386	022572	042737	100100	001164		BIC	#ATA!VV, \$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4387	022600	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B.
4388	022606	016037	000012	001172		MOV	RMD51(RO), \$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.
4389	022614	042737	024001	001172		BIC	#PIP!WRL!OM, \$TMP3		;CLEAR DONT CARES
4390	022622	013737	001172	001166		MOV	\$TMP3, \$TMP1		;COPY IT INTO 'TMP1'
4391	022630	042737	100100	001166		BIC	#ATA!VV, \$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4392	022636	023737	001164	001166		CMP	\$TMP0, \$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4393	022644	001006				BNE	74\$;BR IF NOT
4394	022646	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4395	022652	001045				BNE	76\$;BR IF NOT
4396	022654	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4397	022656	000137	023056			JMP	78\$;BYPASS THE REST OF THE CHECKS
4398	022662	013737	001170	001126	74\$:	MOV	\$TMP2, \$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4399	022670	013737	001226	001240		MOV	PORTB, PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4400	022676	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B.
4401	022704	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
4402	022710	001414				BEQ	75\$;BR IF ZERO
4403	022712	013737	001224	001240		MOV	PORTA, PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4404	022720	013737	001172	001126		MOV	\$TMP3, \$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
4405	022726	113760	001224	000010		MOVB	PORTA, RMCS2(RO)		;SELECT PORT A.
4406	022734	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
4407	022740	001012				BNE	76\$;BR IF NOT
4408	022742	012737	177777	001254	75\$:	MOV	#-1, RELERR		;SET 'RELEASE ERROR' INDICATOR
4409	022750	012760	000011	000000		MOV	#11, RMCS1(RO)		;CLEAR THE DRIVE
4410	022756	012760	000013	000000		MOV	#13, RMCS1(RO)		;RELEASE THE DRIVE
4411	022764	104026				ERROR	26		;TYPE ERROR MESSAGE 26
4412	022766	013737	001170	001126	76\$:	MOV	\$TMP2, \$BDDAT		;LOOK FOR BIT FAILURES WHEN RMD51 READ
4413	022774	013737	001224	001240		MOV	PORTA, PTNBR		;CHANGE PORT NUMBER
4414	023002	042737	100000	001126		BIC	#ATA, \$BDDAT		;DON'T CHECK THE ATTN BIT
4415	023010	023737	001124	001126		CMP	\$GDDAT, \$BDDAT		;ALL BITS OK ?
4416	023016	001401				BEQ	77\$;BR IF OK FROM PORT A.
4417	023020	104007				ERROR	7		;REPORT ERROR
4418	023022	013737	001172	001126	77\$:	MOV	\$TMP3, \$BDDAT		;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
4419	023030	013737	001226	001240		MOV	PORTB, PTNBR		;CHANGE PORT NUMBER
4420	023036	042737	100000	001126		BIC	#ATA, \$BDDAT		;DON'T CHECK THE ATTN BIT
4421	023044	023737	001124	001126		CMP	\$GDDAT, \$BDDAT		;SEE IF READ OK FROM PORT B.

```

4422 023052 001401
4423 023054 104007
4424 023056 000240
4425 023060 000004
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440 023062
4441 023062 005737 001300
4442 023066 001406
4443 023070 100002
4444 023072 000137 002676
4445 023076 012737 177777 001300
4446 023104 112737 000016 001102
4447 023112 012737 023134 001106
4448 023120 012737 023134 001110
4449 023126 012737 000031 001176
4450 023134 012706 001100
4451
4452
4453
4454 023140 113760 001224 000010
4455 023146 005060 000012
4456 023152 012760 000011 000000
4457 023160 012760 000013 000000
4458 023166 113760 001226 000010
4459 023174 005060 000012
4460 023200 012760 000011 000000
4461 023206 012760 000013 000000
4462
4463
4464
4465
4466
4467 023214 113760 001224 000010
4468 023222 013737 001224 001242
4469 023230 005760 000000
4470 023234 113760 001226 000010
4471 023242 013737 001226 001240
4472 023250 013737 001226 001244
4473 023256 016037 000012 001126
4474 023264 010037 001122
4475 023270 062737 000012 001122
4476 023276 005037 001124
4477 023302 023737 001124 001126

```

```

BEQ 785 ;BR IF OK
ERROR 7 ;REPORT ERROR
785: NOP
15: SCOPE ;LOOP ?

*****
*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.
*
*****
†ST16:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #16,$STNM ;TEST NUMBER
MOV #TEST16,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST16,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25. ITERATIONS
TEST16: MOV #STACK,$SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
CLR RMDS1(RO) ;SEIZE THE DRIVE
MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(RO) ;RELEASE THE DRIVE

*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RMCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RMCS1(RO) ;READ RMCS1
MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RMDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BODADR ;R#11 BASE ADDRESS
ADD #RMDS1,$BODADR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO

```

H07

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T16

MACY11 30(1046) 01-AUG-77 11:02 PAGE 85
SEIZE 'A' BY RMCS1 TEST

SEQ 0087

```

4478 023310 001403 BEQ 64$ ;BR IF IT IS
4479 023312 104004 ERROR 4 ;REPORT THE ERROR
4480 023314 000137 024004 JMP 1$ ;BYPASS REST OF THE SUBTEST
4481 023320 64$:
4482 023320 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4483 023326 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4484 023334 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4485 023342 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
4486 023350 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
4487 023356 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
4488 023364 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
4489 023370 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
4490 023376 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
4491 023404 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4492 023412 001401 BEQ 65$ ;BR IF THEY ARE
4493 023414 104005 ERROR 5 ;REPORT THE ERROR
4494 023416 000240 65$: NOP
4495
4496 ;;*****
4497
4498 ;RELEASE THE DRIVE FROM PORT A
4499
4500 023420 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4501 023426 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4502 023434 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4503
4504 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4505
4506 023442 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4507 023446 012737 000012 001122 MOV #RMDS1, $BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4508 023454 060037 001122 ADD RO, $BDDADR ;ADD THE I/O BASE ADDRESS
4509 023460 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
4510 023466 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
4511 023474 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4512 023502 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
4513 023510 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
4514 023516 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4515 023524 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
4516 023532 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4517 023540 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
4518 023546 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
4519 023554 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4520 023562 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4521 023570 001006 BNE 66$ ;BR IF NOT
4522 023572 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4523 023576 001045 BNE 68$ ;BR IF NOT
4524 023600 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4525 023602 000137 024002 JMP 70$ ;BYPASS THE REST OF THE CHECKS
4526 023606 013737 001170 001126 66$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4527 023614 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4528 023622 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
4529 023630 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4530 023634 001414 BEQ 67$ ;BR IF ZERO
4531 023636 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4532 023644 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4533 023652 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.

```

```

4534 023660 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
4535 023664 001012                    BNE      68$           ;BR IF NOT
4536 023666 012737 177777 001254 67$:  MOV     #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
4537 023674 012760 000011 000000    MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
4538 023702 012760 000013 000000    MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
4539 023710 104026                    ERROR   26            ;TYPE ERROR MESSAGE 26
4540 023712 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4541 023720 013737 001224 001240    MOV     PORTA,PTNCR   ;CHANGE PORT NUMBER
4542 023726 042737 100000 001126    BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
4543 023734 023737 001124 001126    CMP     $GDDAT,$BDDAT ;ALL BITS OK
4544 023742 001401                    BEQ     69$           ;BR IF OK FROM PORT A.
4545 023744 104007                    ERROR   7            ;REPORT ERROR
4546 023746 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT  ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4547 023754 013737 001226 001240    MOV     PORTB,PTNBR   ;CHANGE PORT NUMBER
4548 023762 042737 100000 001126    BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
4549 023770 023737 001124 001126    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4550 023776 001401                    BEQ     70$           ;BR IF OK
4551 024000 104007                    ERROR   7            ;REPORT ERROR
4552 024002 000240                    NOP
4553 024004 000004                    1$:     SCOPE          ;LOOP ?

```

```

4554
4555 ;*****
4556 ;*TEST 17 SEIZE 'B' BY RMCS1 TEST
4557 ;*
4558 ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
4559 ;*IF THE DRIVE IS IN NEUTRAL.
4560 ;* A. READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT
4561 ;* THE DRIVE IS SEIZED.
4562 ;*
4563 ;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
4564 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4565 ;*
4566 ;*****

```

```

4567 024006          †ST17:
4568 024006 005737 001300          TST     KYBCTL        ;PERFORMING ONLY SINGLE TESTS ?
4569 024012 001406                    BEQ     2$            ;BR IF NOT
4570 024014 100002                    BPL     1$            ;BR IF JUST ENTERED TEST
4571 024016 000137 002676          JMP     EXEC          ;RETURN & GET NEXT TEST NUMBER
4572 024022 012737 177777 001300 1$:  MOV     #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
4573 024030 112737 000017 001102 2$:  MOVB   #17,$STNM     ;TEST NUMBER
4574 024036 012737 024060 001106    MOV     #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
4575 024044 012737 024060 001110    MOV     #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4576 024052 012737 000031 001176    MOV     #25,$TIMES    ;DO 25 ITERATIONS
4577 024060 012706 001100    TEST17: MOV    #STACK,SP   ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

4581 024064 113760 001224 000010    MOVB   PORTA,RMCS2(RO) ;SELECT PORT #A
4582 024072 005060 000012                    CLR     RMDS1(RO)     ;SEIZE THE DRIVE
4583 024076 012760 000011 000000    MOV     #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4584 024104 012760 000013 000000    MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
4585 024112 113760 001226 000010    MOVB   PORTB,RMCS2(RO) ;SELECT PORT #B
4586 024120 005060 000012                    CLR     RMDS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
4587 024124 012760 000011 000000    MOV     #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4588 024132 012760 000013 000000    MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
4589

```



```

4590 ;*****
4591
4592 ;SEIZE THE DRIVE THROUGH PORT B
4593
4594 024140 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4595 024146 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
4596 024154 005760 000000 TST RMCS1(RO) ;READ RMCS1
4597 024160 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4598 024166 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4599 024174 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
4600 024202 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
4601 024210 010037 001122 MOV RO, $BDAOR ;R11 BASE ADDRESS
4602 024214 062737 000012 001122 ADD #RMDS1, $BDAOR ;GENERATE BAD REGISTER ADDRESS
4603 024222 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
4604 024226 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
4605 024234 001403 BEQ 64$ ;BR IF IT IS
4606 024236 104004 ERROR 4 ;REPORT THE ERROR
4607 02424J 000137 024730 JMP 1$ ;BYPASS REST OF THE SUBTEST
4608 024244
4609 024244 113760 001226 000010 64$: MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4610 024252 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4611 024260 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4612 024266 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
4613 024274 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
4614 024302 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
4615 024310 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
4616 024314 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
4617 024322 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
4618 024330 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4619 024336 001401 BEQ 65$ ;BR IF THEY ARE
4620 024340 104005 ERROR 5 ;REPORT THE ERROR
4621 024342 000240 65$: NOP
4622
4623 ;*****
4624
4625 ;RELEASE THE DRIVE FROM PORT B
4626
4627 024344 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4628 024352 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4629 024360 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4630
4631 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4632
4633 024366 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4634 024372 012737 000012 001122 MOV #RMDS1, $BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4635 024400 060037 001122 ADD RO, $BDAOR ;ADD THE I/O BASE ADDRESS
4636 024404 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
4637 024412 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4638 024420 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4639 024426 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
4640 024434 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
4641 024442 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4642 024450 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4643 024456 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4644 024464 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
4645 024472 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'

```

```

4646 024500 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4647 024506 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4648 024514 001006 BNE 66$ ;BR IF NOT
4649 024516 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4650 024522 001045 BNE 68$ ;BR IF NOT
4651 024524 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4652 024526 000137 024726 JMP 70$ ;BYPASS THE REST OF THE CHECKS
4653 024532 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4654 024540 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4655 024546 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
4656 024554 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4657 024560 001414 BEQ 67$ ;BR IF ZERO
4658 024562 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4659 024570 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4660 024576 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
4661 024604 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4662 024610 001012 BNE 68$ ;BR IF NOT
4663 024612 012737 177777 001254 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4664 024620 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
4665 024626 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
4666 024634 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
4667 024636 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4668 024644 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
4669 024652 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4670 024660 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
4671 024666 001401 BEQ 69$ ;BR IF OK FROM PORT A.
4672 024670 104007 ERROR 7 ;REPORT ERROR
4673 024672 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4674 024700 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4675 024706 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4676 024714 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4677 024722 001401 BEQ 70$ ;BR IF OK
4678 024724 104007 ERROR 7 ;REPORT ERROR
4679 024726 000240 70$: NOP
4680 024730 000004 1$: SCOPE ;LOOP ?

```

```

4681
4682
4683 ;*****
4684 ;TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST
4685 ;
4686 ;VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
4687 ;REQUEST' IF THE DRIVE IS SEIZED.
4688 ;
4689 ; A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1. VERIFY THAT
4690 ; THE DRIVE HAS BEEN SEIZED.
4691 ;
4692 ; B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT
4693 ; SET.
4694 ;
4695 ; C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
4696 ; RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4697 ;
4698 ;*****
4699 †ST20:
4700 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
4701 BEQ 2$ ;BR IF NOT

```

L07

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T20

MACY11 30(1046) 01-AUG-77 11:02 PAGE 89
PORT 'A' INHIBIT SEIZE BY RMCS1 TEST

SEQ 0091

```

4702 024740 100002          BPL      1$          ;BR IF JUST ENTERED TEST
4703 024742 000137 002676    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
4704 024746 012737 177777 001300 1$:      MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
4705 024754 112737 000020 001102 2$:      MOVVB   #20,$TSTNM   ;TEST NUMBER
4706 024762 012737 025004 001106    MOV      #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
4707 024770 012737 025004 001110    MOV      #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4708 024776 012737 000031 001176    MOV      #25,$TIMES    ;DO 25. ITERATIONS
4709 025004 012706 001100    TEST20: MOV     #STACK,SP ;LOAD THE STACK POINTER
4710
4711                      ;CLEAR ATTENTION BITS FOR BOTH PORTS
4712
4713 025010 113760 001224 000010    MOVVB   PORTA,RMCS2(RO) ;SELECT PORT #A
4714 025016 005060 000012          CLR      RMCS1(RO)     ;SEIZE THE DRIVE
4715 025022 012760 000011 000000    MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4716 025030 012760 000013 000000    MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
4717 025036 113760 001226 000010    MOVVB   PORTB,RMCS2(RO) ;SELECT PORT #B
4718 025044 005060 000012          CLR      RMCS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
4719 025050 012760 000011 000000    MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4720 025056 012760 000013 000000    MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
4721
4722                      ;*****
4723
4724                      ;SEIZE THE DRIVE THROUGH PORT B
4725
4726 025064 113760 001226 000010    MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B
4727 025072 013737 001226 001242    MOV      PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
4728 025100 005760 000000          TST      RMCS1(RO)     ;READ RMCS1
4729 025104 113760 001224 000010    MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A
4730 025112 013737 001224 001240    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4731 025120 013737 001224 001244    MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
4732 025126 016037 000012 001126    MOV      RMCS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
4733 025134 010037 001122          MOV      RO,$BADDR     ;R#11 BASE ADDRESS
4734 025140 062737 000012 001122    ADD      #RMCS1,$BADDR  ;GENERATE BAD REGISTER ADDRESS
4735 025146 005037 001124          CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
4736 025152 023737 001124 001126    CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
4737 025160 001403          BEQ      64$          ;BR IF IT IS
4738 025162 104004          ERROR   4             ;REPORT THE ERROR
4739 025164 000137 025776          JMP      1$           ;BYPASS REST OF THE SUBTEST
4740
4741 025170 113760 001226 000010    64$:  MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B
4742 025176 013737 001226 001240    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4743 025204 016037 000012 001126    MOV      RMCS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4744 025212 042737 020001 001126    BIC      #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
4745 025220 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
4746 025226 013737 001124 001166    MOV      $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
4747 025234 005137 001166          COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
4748 025240 013737 001126 001164    MOV      $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
4749 025246 043737 001166 001164    BIC      $TMP1,$TMP0   ;CLEAR UNWANTED BITS
4750 025254 023737 001124 001164    CMP      $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
4751 025262 001401          BEQ      65$          ;BR IF THEY ARE
4752 025264 104005          ERROR   5             ;REPORT THE ERROR
4753
4754 025270 113760 001224 000010    65$:  NOP
4755 025276 013737 001224 001240    MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A
4756          MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4757
;*****

```

```

4758 ;READ RMCS1 THROUGH PORT A - TRY TO SET PORT REQUEST
4759
4750 025304 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4761 025310 016037 000000 001126 MOV RMCS1(RO), $BODAT ;GET CONTENTS OF RMCS1
4762 025316 012737 000000 001122 MOV #RMCS1, $BODADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4763 025324 060037 001122 ADD RO, $BODADR ;ADD RM11 BASE ADDRESS
4764 025330 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
4765 025334 013737 001126 001164 MOV $BODAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4766 025342 042737 173700 001164 BIC #1C4077, $TMP0 ;SAVE SPECIFIED BITS
4767 025350 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
4768 025356 001414 BEQ 665 ;BR IF OK
4769 025360 013737 001126 001174 MOV $BODAT, $TMP4 ;COPY 'BAD DATA'
4770 025366 042737 004077 001174 BIC #4077, $TMP4 ;CLEAR THE MASKED BITS
4771 025374 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4772 025402 104010 ERROR 10 ;REPORT THE ERROR
4773 025404 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4774 025410 000240 665: NOP
4775
4776 ;*****
4777 ;DRIVE SHOULD RETURN TO NEUTRAL
4778
4779
4780 ;RELEASE THE DRIVE FROM PORT B
4781
4782 025412 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4783 025420 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4784 025426 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4785
4786 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4787
4788 025434 005037 001254 CLR RELEARR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4789 025440 012737 000012 001122 MOV #RMS1, $BODADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
4790 025446 060037 001122 ADD RO, $BODADR ;ADD THE I/O BASE ADDRESS
4791 025452 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
4792 025460 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
4793 025466 016037 000012 001170 MOV RMS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4794 025474 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
4795 025502 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
4796 025510 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4797 025516 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
4798 025524 016037 000012 001172 MOV RMS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4799 025532 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
4800 025540 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
4801 025546 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4802 025554 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4803 025562 001006 BNE 685 ;BR IF NOT
4804 025564 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4805 025570 001045 BNE 705 ;BR IF NOT
4806 025572 104046 ERROR 16 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4807 025574 000137 025774 JMP 725 ;BYPASS THE REST OF THE CHECKS
4808 025600 013737 001170 001126 685: MOV $TMP2, $BODAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4809 025606 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4810 025614 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
4811 025622 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4812 025626 001414 BEQ 695 ;BR IF ZERO
4813 025630 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    
```

```

4814 025636 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4815 025644 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
4816 025652 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4817 025656 001012 BNE 70$ ;BR IF NOT
4818 025660 012737 177777 001254 69$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
4819 025666 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
4820 025674 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
4821 025702 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
4822 025704 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR 9IT FAILURES WHEN RMDS1 READ
4823 025712 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
4824 025720 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4825 025726 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
4826 025734 001401 BEQ 71$ ;BR IF OK FROM PORT A.
4827 025736 104007 ERROR 7 ;REPORT ERROR
4828 025740 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4829 025746 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4830 025754 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4831 025762 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4832 025770 001401 BEQ 72$ ;BR IF OK
4833 025772 104007 ERROR 7 ;REPORT ERROR
4834 025774 000240 72$: NOP
4835 025776 000004 1$: SCOPE ;LOOP ?

```

```

4836
4837 ;*****
4838 ;*TEST 21 PORT 'B' INHIBIT SEIZE BY RMCS1 TEST
4839 ;*
4840 ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
4841 ;* REQUEST' IF THE DRIVE IS SEIZED.
4842 ;*
4843 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1. VERIFY THAT
4844 ;* THE DRIVE HAS BEEN SEIZED.
4845 ;*
4846 ;* B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT
4847 ;* SET.
4848 ;*
4849 ;* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
4850 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4851 ;*
4852 ;*****

```

```

4853 026000
4854 026000 005737 001300 †ST21: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
4855 026004 001406 BEQ 2$ ;BR IF NOT
4856 026006 100002 BPL 1$ ;BR IF JUST ENTERED TEST
4857 026010 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
4858 026014 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
4859 026022 112737 000021 001102 2$: MOVB #21,$TSTNM ;TEST NUMBER
4860 026030 012737 026052 001106 MOV #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS
4861 026036 012737 026052 001110 MOV #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4862 026044 012737 000031 001176 MOV #25,$TIMES ;DO 25. ITERATIONS
4863 026052 012706 001100 TEST21: MOV #STACK,SP ;LOAD THE STACK POINTER
4864
4865 ;CLEAR ATTENTION BITS FOR BOTH PORTS
4866
4867 026056 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
4868 026064 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
4869 026070 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR

```

```

4870 026076 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
4871 026104 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT #B
4872 026112 005060 000012      CLR      RMCS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
4873 026116 012760 000011 000000      MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4874 026124 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
4875
4876 ;*****
4877
4878 ;SEIZE THE DRIVE THROUGH PORT A
4879
4880 026132 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
4881 026140 013737 001224 001242      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
4882 026146 005760 000000      TST      RMCS1(RO) ;READ RMCS1
4883 026152 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
4884 026160 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4885 026166 013737 001226 001244      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
4886 026174 016037 000012 001126      MOV      RMCS1(RO),%BDDAT ;SEE IF DRIVE SEIZED BY PORT A
4887 026202 010037 001122      MOV      RO,%BDDADR ;RH11 BASE ADDRESS
4888 026206 062737 000012 001122      ADD      #RMCS1,%BDDADR ;GENERATE BAD REGISTER ADDRESS
4889 026214 005037 001124      CLR      %GDDAT ;REGISTER SHOULD BE ZERO
4890 026220 023737 001124 001126      CMP      %GDDAT,%BDDAT ;IS THE REGISTER ZERO
4891 026226 001403      BEQ      64$ ;BR IF IT IS
4892 026230 104004      ERROR   4 ;REPORT THE ERROR
4893 026232 000137 027044      JMP      1$ ;BYPASS REST OF THE SUBTEST
4894 026236
4895 026236 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
4896 026244 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4897 026252 016037 000012 001126      MOV      RMCS1(RO),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4898 026250 042737 020001 001126      BIC      #OM!PIP,%BDDAT ;CLEAR DONT CARE BITS
4899 026266 012737 011700 001124      MOV      #MOL!PGH!DPR!DRY!VV,%GDDAT ;EXPECTED STATUS
4900 026274 013737 001124 001166      MOV      %GDDAT,%TMP1 ;USE GOOD DATA AS A MASK
4901 026302 005137 001166      COM      %TMP1 ;COMPLEMENT THE EXPECTED STATUS
4902 026306 013737 001126 001164      MOV      %BDDAT,%TMP0 ;SAVE THE ACTUAL STATUS
4903 026314 043737 001166 001164      BIC      %TMP1,%TMP0 ;CLEAR UNWANTED BITS
4904 026322 023737 001124 001164      CMP      %GDDAT,%TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4905 026330 001401      BEQ      65$ ;BR IF THEY ARE
4906 026332 104005      ERROR   5 ;REPORT THE ERROR
4907 026334 000240      NOP
4908 026336 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
4909 026344 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4910
4911 ;*****
4912 ;READ RMCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
4913
4914 026352 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4915 026356 016037 000000 001126      MOV      RMCS1(RO),%BDDAT ;GET CONTENTS OF RMCS1
4916 026364 012737 000000 001122      MOV      #RMCS1,%BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4917 026372 060037 001122      ADD      RO,%BDDADR ;ADD RH11 BASE ADDRESS
4918 026376 005037 001124      CLR      %GDDAT ;WHAT REGISTER SHOULD BE
4919 026402 013737 001126 001164      MOV      %BDDAT,%TMP0 ;MOVE REGISTER CONTENTS TO '%TMP0'
4920 026410 042737 173700 001164      BIC      #1C4077,%TMP0 ;SAVE SPECIFIED BITS
4921 026416 023737 001124 001164      CMP      %GDDAT,%TMP0 ;COMPARE THE BITS
4922 026424 001414      BEQ      66$ ;BR IF OK
4923 026426 013737 001126 001174      MOV      %BDDAT,%TMP4 ;COPY 'BAD DATA'
4924 026434 042737 004077 001174      BIC      #4077,%TMP4 ;CLEAR THE MASKED BITS
4925 026442 053737 001174 001124      BIS      %TMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

```

4926 026450 104010          ERROR 10          ;REPORT THE ERROR
4927 026452 005137 001250  COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
4928 026456 000240          66$:  NOP
4929
4930 ;:*****
4931 ;DRIVE SHOULD RETURN TO NEUTRAL
4932
4933
4934 ;RELEASE THE DRIVE FROM PORT A
4935
4936 026460 113760 001224 000010  MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
4937 026466 013737 001224 001240  MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4938 026474 012760 000013 000000  MOV   #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4939
4940 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4941
4942 026502 005037 001254          CLR   RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
4943 026506 012737 000012 001122  MOV   #RMDS1,$BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4944 026514 060037 001122          ADD   RO,$BDDADR  ;ADD THE I/O BASE ADDRESS
4945 026520 012737 011700 001124  MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4946 026526 113760 001224 000010  MOVB  PORTA,RMCS2(RO) ;SELECT PORT A.
4947 026534 016037 000012 001170  MOV   RMDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4948 026542 042737 024001 001170  BIC   #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
4949 026550 013737 001170 001164  MOV   STMP2,STMP0 ;COPY IT INTO 'STMP0'
4950 026556 042737 100100 001164  BIC   #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4951 026564 113760 001226 000010  MOVB  PORTB,RMCS2(RO) ;SELECT PORT B.
4952 026572 016037 000012 001172  MOV   RMDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4953 026600 042737 024001 001172  BIC   #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
4954 026606 013737 001172 001166  MOV   STMP3,STMP1 ;COPY IT INTO 'STMP1'
4955 026614 042737 100100 001166  BIC   #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4956 026622 023737 001164 001166  CMP   STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4957 026630 001006          BNE   68$        ;BR IF NOT
4958 026632 005737 001164          TST   STMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4959 026636 001045          BNE   70$        ;BR IF NOT
4960 026640 104046          ERROR 46        ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4961 026642 000137 027042          JMP   72$        ;BYPASS THE REST OF THE CHECKS
4962 026646 013737 001170 001126 68$: MOV   STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4963 026654 013737 001226 001240  MOV   PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4964 026662 113760 001226 000010  MOVB  PORTB,RMCS2(RO) ;SELECT PORT B.
4965 026670 005737 001164          TST   STMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
4966 026674 001414          BEQ   69$        ;BR IF ZERO
4967 026676 013737 001224 001240  MOV   PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4968 026704 013737 001172 001126  MOV   STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4969 026712 113760 001224 000010  MOVB  PORTA,RMCS2(RO) ;SELECT PORT A.
4970 026720 005737 001166          TST   STMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
4971 026724 001012          BNE   70$        ;BR IF NOT
4972 026726 012737 177777 001254 69$: MOV   #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4973 026734 012760 000011 000000  MOV   #11,RMCS1(RO) ;CLEAR THE DRIVE
4974 026742 012760 000013 000000  MOV   #13,RMCS1(RO) ;RELEASE THE DRIVE
4975 026750 104026          ERROR 26        ;TYPE ERROR MESSAGE 26
4976 026752 013737 001170 001126 70$: MOV   STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4977 026760 013737 001224 001240  MOV   PORTA,PTNBR ;CHANGE PORT NUMBER
4978 026766 042737 100000 001126  BIC   #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4979 026774 023737 001124 001126  CMP   $GDDAT,$BDDAT ;ALL BITS OK ?
4980 027002 001401          BEQ   71$        ;BR IF OK FROM PORT A.
4981 027004 104007          ERROR 7         ;REPORT ERROR

```

```

4982 027006 013737 001172 001126 71$: MOV STMP3, $BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4983 027014 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
4984 027022 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
4985 027030 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
4986 027036 001401 BEQ 72$ ;BR IF OK
4987 027040 104007 ERROR 7 ;REPORT ERROR
4988 027042 000240 72$: NOP
4989 027044 000004 1$: SCOPE ;LOOP ?
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006

```

```

*****
;TEST 22 SEIZE BY RMAS TEST
*****

```

```

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

```

```

*****
;TEST 22:
*****

```

```

5007 027046 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5008 027046 001406 BEQ 2$ ;BR IF NOT
5009 027052 001406 BPL 1$ ;BR IF JUST ENTERED TEST
5010 027054 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5011 027056 000137 002676 1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
5012 027062 012737 177777 001300 2$: MOVB #22, $STNM ;TEST NUMBER
5013 027070 112737 000022 001102 MOV #TEST22, $LPADR ;LOAD LOOP ON TEST ADDRESS
5014 027076 012737 027120 001106 MOV #TEST22, $LPERR ;LOAD LOOP ON ERROR ADDRESS
5015 027104 012737 027120 001110 MOV #25, $TIMES ;DO 25. ITERATIONS
5016 027112 012737 000031 001176 TEST22: MOV #STACK, SP ;LOAD THE STACK POINTER
5017 027120 012706 001100

```

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

```

5021 027124 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT #A
5022 027132 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
5023 027136 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
5024 027144 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5025 027152 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT #B
5026 027160 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
5027 027164 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
5028 027172 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5029
5030

```

```

*****
;SELECT DRIVE OTHER THAN THAT BEING TESTED
*****

```

```

5033 027200 113760 001230 000010 MOVB PORTC, RMCS2(RO) ;SELECT DRIVE NOT BEING TESTED
5034 027206 013737 001224 001242 MOV PORTA, SEIZPT ;'SEIZED' PORT ADDRESS
5035
5036
5037

```

```

*****
;WRITE THE DRIVE'S ATTENTION BIT
*****

```



```

5038
5039 027214 013760 001236 000016      MOV   ASR1,RMAS(RO) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
5040 027222 113760 001224 000010      MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
5041 027230 013737 001224 001240      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5042
5043 ;:*****
5044 ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
5045
5046 027236 005760 000012      TST   RMDS1(RO) ;SEE THE REGISTER THROUGH PORT A ?
5047 027242 001014          BNE   1$ ;BR IF YES
5048 027244 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
5049 027252 013737 001226 001240      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5050 027260 005760 000012      TST   RMDS1(RO) ;SEE REGISTER THROUGH PORT B ?
5051 027264 001021          BNE   2$ ;BR IF YES
5052 027266 104042          ERROR 42 ;DRIVE NOT IN NEUTRAL OR SEIZED
5053 027270 000137 031120          JMP   4$ ;BYPASS REST OF TEST
5054 027274
5055 027274 113760 001226 000010      1$: MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
5056 027302 013737 001226 001240      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5057 027310 005760 000012      TST   RMDS1(RO) ;REGISTER SHOULD BE ZERO THROUGH PORT B
5058 027314 001002          BNE   .+6 ;BR IF STATUS REG IS NOT ZERO
5059 027316 000137 030220          JMP   3$ ;STATUS REG IS ZERO
5060 027322 104043          ERROR 43 ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
5061 027324 000137 031120          JMP   4$ ;BYPASS REST OF TEST
5062
5063 ;:*****
5064 ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
5065
5066 027330
5067 027330 005037 001250          2$: CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5068 027334 016037 000012 001126      MOV   RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
5069 027342 012737 000012 001122      MOV   #RMDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5070 027350 060037 001122          ADD   RO,SBADR ;ADD RH11 BASE ADDRESS
5071 027354 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,SGDDAT ;WHAT REGISTER SHOULD BE
5072 027362 013737 001126 001164      MOV   SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5073 027370 042737 106077 001164      BIC   #+C71700,$TMP0 ;SAVE SPECIFIED BITS
5074 027376 023737 001124 001164      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
5075 027404 001414          BEQ   64$ ;BR IF OK
5076 027406 013737 001126 001174      MOV   SBDDAT,$TMP4 ;COPY 'BAD DATA'
5077 027414 042737 071700 001174      BIC   #71700,$TMP4 ;CLEAR THE MASKED BITS
5078 027422 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5079 027430 104010          ERROR 10 ;REPORT THE ERROR
5080 027432 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5081 027436 000240          64$: NOP
5082 027440 013737 001226 001242      MOV   PORTB,SEIZPT ;ADDRESS FOR ERROR MESSAGE
5083 027446 013737 001224 001244      MOV   PORTA,OPPRT ;SAME AS ABOVE
5084
5085 ;RELEASE THE DRIVE FROM PORT B
5086
5087 027454 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
5088 027462 013737 001226 001240      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5089 027470 012760 000013 000000      MOV   #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5090
5091 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5092
5093 027476 005037 001254          CLR   RELERR ;CLEAR 'RELEASE ERROR' INDICATOR

```

F08

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T22

MACY11 30(1046) 01-AUG-77 11:02 PAGE 96
 SEIZE BY RMAS TEST

SEQ 0098

5094	027502	012737	111700	001124		MOV	#ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
5095	027510	012737	000012	001122		MOV	#RMS1,\$B0ADR	;REGISTER ADDRESS INCREMENT
5096	027516	060037	001122			ADD	RO,\$B0ADR	;REGISTER BASE ADDRESS FOR TYPEOUT
5097	027522	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
5098	027530	013737	001224	001240		MOV	PORTA, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5099	027536	016037	000012	001164		MOV	RMS1(RO), \$TMP0	;READ STATUS REGISTER FROM PORT A
5100	027544	113760	001226	000010		MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
5101	027552	013737	001226	001240		MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5102	027560	016037	000012	001126		MOV	RMS1(RO), \$BDDAT	;DRIVE STATUS FROM PORT B
5103	027566	001404				BEQ	66\$;BR IF STATUS FROM PORT B ZERO
5104	027570	005737	001164			TST	\$TMP0	;IS STATUS FROM PORT A ZERO ?
5105	027574	061401				BEQ	66\$;BR IF ZERO
5106	027576	104044				ERROR	44	;REPORT DRIVE NOT SEIZED BY PORT A
5107	027600	013737	001164	001126	66\$:	MOV	\$TMP0, \$BDDAT	;CHECK STATUS FROM PORT A
5108	027606	013737	001224	001240		MOV	PORTA, PTNBR	;CHANGE PORT ADDRESS FOR TYPEOUT
5109	027614	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	;COMPARE WITH CONSTANT
5110	027622	001401				BEQ	67\$;BR IF OK
5111	027624	104027				ERROR	27	;REPORT REGISTER ERROR
5112	027626	000240			67\$:	NOP		
5113								
5114								;RELEASE THE DRIVE FROM PORT A
5115								
5116	027630	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
5117	027636	013737	001224	001240		MOV	PORTA, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5119	027644	012760	000013	000000		MOV	#13, RMS1(RO)	;ISSUE RELEASE THROUGH PORT A
5119								
5120								;VERIFY THAT THE DRIVE IS IN NEUTRAL
5121								
5122	027652	005037	001254			CLR	RELER	;CLEAR THE 'RELEASE ERROR' INDICATOR
5123	027656	012737	000012	001122		MOV	#RMS1,\$B0ADR	;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
5124	027664	060037	001122			ADD	RO,\$B0ADR	;ADD THE I/O BASE ADDRESS
5125	027670	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
5126	027676	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A.
5127	027704	016037	000012	001170		MOV	RMS1(RO), \$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
5128	027712	042737	024001	001170		BIC	#PIP!WRL!OM,\$TMP2	;CLEAR DONT CARES
5129	027720	013737	001170	001164		MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
5130	027726	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
5131	027734	113760	001226	000010		MOVB	PORTB, RMCS2(RO)	;SELECT PORT B.
5132	027742	016037	000012	001172		MOV	RMS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
5133	027750	042737	024001	001172		BIC	#PIP!WRL!OM,\$TMP3	;CLEAR DONT CARES
5134	027756	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
5135	027764	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
5136	027772	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5137	030000	001006				BNE	68\$;BR IF NOT
5138	030002	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5139	030006	001045				BNE	70\$;BR IF NOT
5140	030010	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5141	030012	000137	030212			JMP	72\$;BYPASS THE REST OF THE CHECKS
5142	030016	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5143	030024	013737	001226	001240		MOV	PORTB, PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5144	030032	113760	001226	000010		MOVB	PORTB, RMCS2(RO)	;SELECT PORT B.
5145	030040	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
5146	030044	001414				BEQ	69\$;BR IF ZERO
5147	030046	013737	001224	001240		MOV	PORTA, PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5148	030054	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
5149	030062	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A.

```

5150 030070 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
5151 030074 001012                    BNE      70$           ;BR IF NOT
5152 030076 012737 177777 001254 69$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
5153 030104 012760 000011 000000    MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
5154 030112 012760 000013 000000    MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5155 030120 104026                    ERROR    26           ;TYPE ERROR MESSAGE 26
5156 030122 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5157 030130 013737 001224 001240    MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
5158 030136 042737 100000 001126    BIC      #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
5159 030144 023737 001124 001126    CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
5160 030152 001401                    BEQ      71$           ;BR IF OK FROM PORT A.
5161 030154 104007                    ERROR    7            ;REPORT ERROR
5162 030156 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT   ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5163 030164 013737 001226 001240    MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
5164 030172 042737 100000 001126    BIC      #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
5165 030200 023737 001124 001126    CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5166 030206 001401                    BEQ      72$           ;BR IF OK
5167 030210 104007                    ERROR    7            ;REPORT ERROR
5168 030212 000240                    NOP
5169 030214 000137 031120          JMP      4$
5170
5171 ;*****
5172 ;THE DRIVE IS SEIZED BY PORT A. VERIFY THAT PORT B HAS PORT REQUEST SET
5173
5174 030220          3$:
5175 030220 113760 001224 000010    MOV      PORTA,RMCS2(RO) ;SELECT PORT A
5176 030226 013737 001224 001240    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5177 030234 005037 001250          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
5178 030240 016037 000012 001126    MOV      RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
5179 030246 012737 000012 001122    MOV      #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5180 030254 060037 001122          ADD      RO,$BDDADR    ;ADD RHI1 BASE ADDRESS
5181 030260 012737 011700 001124    MOV      #MCL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
5182 030266 013737 001126 001164    MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO 'TMP0'
5183 030274 042737 106077 001164    BIC      #1C71700,$TMP0 ;SAVE SPECIFIED BITS
5184 030302 023737 001124 001164    CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
5185 030310 001414                    BEQ      73$           ;BR IF OK
5186 030312 013737 001126 001174    MOV      $BDDAT,$TMP4  ;COPY 'BAD DATA'
5187 030320 042737 071700 001174    BIC      #71700,$TMP4  ;CLEAR THE MASKED BITS
5188 030326 053737 001174 001124    BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
5189 030334 104010                    ERROR    10           ;REPORT THE ERROR
5190 030336 005137 001250          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
5191 030342 000240                    NOP
5192 030344 013737 001224 001242 73$:  MOV      PORTA,SEIZPT  ;ADDRESS FOR ERROR MESSAGE
5193 030352 013737 001226 001244    MOV      PORTB,OPRPT   ;SAME AS ABOVE
5194
5195 ;RELEASE THE DRIVE FROM PORT A
5196
5197 030360 113760 001224 000010    MOV      PORTA,RMCS2(RO) ;SELECT PORT A
5198 030366 013737 001224 001240    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5199 030374 012760 000013 000000    MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5200
5201 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5202
5203 030402 065037 001254          CLR      RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
5204 030406 012737 111700 001124    MOV      #ATA!MCL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5205 030414 012737 000012 001122    MOV      #RMDS1,$BDDADR ;REGISTER ADDRESS INCREMENT

```

H08

MD-11-DZRMG-A, RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T22

MACY11 30(1046) 01-AUG-77 11:02 PAGE 98
SEIZE BY RMAS TEST

SEQ 0100

```

5206 030422 060037 001122      ADD      RO, $B0ADR      ; REGISTER BASE ADDRESS FOR TYPEOUT
5207 030426 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ; SELECT PORT B
5208 030434 013737 001226 001240  MOV     PORTB, PTNBR     ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5209 030442 016037 000012 001164  MOV     RMD51(RO), $TMP0 ; READ STATUS REGISTER FROM PORT B
5210 030450 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ; SELECT PORT A
5211 030456 013737 001224 001240  MOV     PORTA, PTNBR     ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5212 030464 016037 000012 001126  MOV     RMD51(RO), $B0DAT ; DRIVE STATUS FROM PORT A
5213 030472 001404                BEQ     75$              ; BR IF STATUS FROM PORT A ZERO
5214 030474 005737 001164                TST     $TMP0           ; IS STATUS FROM PORT B ZERO ?
5215 030500 001401                BEQ     75$              ; BR IF ZERO
5216 030502 104044                ERROR   44              ; REPORT DRIVE NOT SEIZED BY PORT B
5217 030504 013737 001164 001126 75$:    MOV     $TMP0, $B0DAT   ; CHECK STATUS FROM PORT B
5218 030512 013737 001226 001240  MOV     PORTB, PTNBR     ; CHANGE PORT ADDRESS FOR TYPEOUT
5219 030520 023737 001124 001126  CMP     $G0DAT, $B0DAT  ; COMPARE WITH CONSTANT
5220 030526 001401                BEQ     76$              ; BR IF OK
5221 030530 104027                ERROR   27              ; REPORT REGISTER ERROR
5222 030532 000240                76$:    NOP
5223
5224                ; RELEASE THE DRIVE FROM PORT B
5225
5226 030534 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ; SELECT PORT B
5227 030542 013737 001226 001240  MOV     PORTB, PTNBR     ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5228 030550 012760 000013 000000  MOV     #13, RMCS1(RO)  ; ISSUE RELEASE THROUGH PORT B
5229
5230                ; VERIFY THAT THE DRIVE IS IN NEUTRAL
5231
5232 030556 005037 001254                CLR     RELERR          ; CLEAR THE 'RELEASE ERROR' INDICATOR
5233 030562 012737 000012 001122  MOV     #RMD51, $B0ADR  ; FORM THE ADDRESS OF RMD51 FOR TYPEOUT
5234 030570 060037 001122                ADD     RO, $B0ADR      ; ADD THE I/O BASE ADDRESS
5235 030574 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV, $G0DAT ; COMPARISON CONSTANT
5236 030602 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ; SELECT PORT A
5237 030610 016037 000012 001170  MOV     RMD51(RO), $TMP2 ; GET THE DRIVE STATUS REGISTER FROM PORT A
5238 030616 042737 024001 001170  BIC     #PIP!WRL!OM, $TMP2 ; CLEAR DONT CARES
5239 030624 013737 001170 001164  MOV     $TMP2, $TMP0    ; COPY IT INTO 'TMP0'
5240 030632 042737 100100 001164  BIC     #ATA!VV, $TMP0   ; CLEAR PORT DEPENDENT BITS FROM THE COPY
5241 030640 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ; SELECT PORT B
5242 030646 016037 000012 001172  MOV     RMD51(RO), $TMP3 ; GET THE DRIVE STATUS REGISTER FROM PORT B
5243 030654 042737 024001 001172  BIC     #PIP!WRL!OM, $TMP3 ; CLEAR DONT CARES
5244 030662 013737 001172 001166  MOV     $TMP3, $TMP1    ; COPY IT INTO 'TMP1'
5245 030670 042737 100100 001166  BIC     #ATA!VV, $TMP1   ; CLEAR PORT DEPENDENT BITS FROM THE COPY
5246 030676 023737 001164 001166  CMP     $TMP0, $TMP1    ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5247 030704 001006                BNE     77$              ; BR IF NOT
5248 030706 005737 001164                TST     $TMP0           ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
5249 030712 001045                BNE     79$              ; BR IF NOT
5250 030714 104046                ERROR   46              ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5251 030716 000137 031116                JMP     81$              ; BYPASS THE REST OF THE CHECKS
5252 030722 013737 001170 001126 77$:    MOV     $TMP2, $B0DAT   ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5253 030730 013737 001226 001240  MOV     PORTB, PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5254 030736 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ; SELECT PORT B
5255 030744 005737 001164                TST     $TMP0           ; SEE IF STATUS EQ 0 FROM PORT A
5256 030750 001414                BEQ     78$              ; BR IF ZERO
5257 030752 013737 001224 001240  MOV     PORTA, PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5258 030760 013737 001172 001126  MOV     $TMP3, $B0DAT   ; 'BAD DATA' FOR ERROR TYPE OUT
5259 030766 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ; SELECT PORT A
5260 030774 005737 001166                TST     $TMP1           ; SEE IF STATUS EQ ZERO FROM PORT B
5261 031000 001012                BNE     79$              ; BR IF NOT

```

```

5262 031002 012737 177777 001254 78$: MOV #1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5263 031010 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
5264 031016 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5265 031024 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5266 031026 013737 001170 001126 79$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5267 031034 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5268 031042 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5269 031050 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
5270 031056 001401 BEQ BOS ;BR IF OK FROM PORT A.
5271 031060 104007 ERROR 7 ;REPORT ERROR
5272 031062 013737 001172 001126 80$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5273 031070 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5274 031076 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5275 031104 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5276 031112 001401 BEQ B15 ;BR IF OK
5277 031114 104007 ERROR 7 ;REPORT ERROR
5278 031116 000240 81$: NOP
5279 031120 000004 4$: SCOPE ;LOOP ?

```

```

5280
5281
5282 ;*****
5283 ;*TEST 23 INHIBIT SEIZE BY RMAS TEST
5284 ;*
5285 ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
5286 ;* THE DRIVE'S ATTENTION BIT.
5287 ;*
5288 ;* A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
5289 ;* BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
5290 ;*
5291 ;* B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
5292 ;*
5293 ;*****

```

```

5294 031122
5295 031122 005737 001300 †ST23: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5296 031126 001406 BEQ 25 ;BR IF NOT
5297 031130 100002 BPL 15 ;BR IF JUST ENTERED TEST
5298 031132 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5299 031136 012737 177777 001300 1$: MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR
5300 031144 112737 000023 001102 2$: MOVB #23,$STNM ;TEST NUMBER
5301 031152 012737 031174 001106 MOV #TEST23,$LPADR ;LOAD LOOP ON TEST ADDRESS
5302 031160 012737 031174 001110 MOV #TEST23,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5303 031166 012737 000031 001176 MOV #25,$TIMES ;DO 25. ITERATIONS
5304 031174 012706 001100 TEST23: MOV #STACK,SP ;LOAD THE STACK POINTER

```

```

5305
5306 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5307
5308 031200 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
5309 031206 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
5310 031212 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5311 031220 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5312 031226 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
5313 031234 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
5314 031240 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5315 031246 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5316 031254 113760 001230 000010 MOVB PORTC,RMCS2(RO) ;SELECT DRIVE NOT BEING TESTED
5317

```

J08

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T23

MACY11 30(1046) 01-AUG-77 11:02 PAGE 100
 INHIBIT SEIZE BY RMAS TEST

SEQ 0102

```

5318 ;*****
5319 ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST
5320
5321 031262 013737 001236 001164      MOV    ASR1,STMP0      ;STORE ATTN BIT FOR PORT A
5322 031270 005137 001164      COM    STMP0          ;COMPLEMENT IT
5323 031274 013760 001164 000016      MOV    STMP0,RMAS(RO) ;WRITE THE ATTN REGISTER
5324
5325 ;*****
5326 ;VERIFY THAT DRIVE REMAINED IN NEUTRAL
5327
5328
5329 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5330
5331 031302 005037 001254      CLR    RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
5332 031306 012737 000012 001.22      MOV    #RMS1,$B0ADR  ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
5333 031314 060037 001122      ADD    RO,$B0ADR     ;ADD THE I/O BASE ADDRESS
5334 031320 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5335 031326 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
5336 031334 016037 000012 001170      MOV    RMS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5337 031342 042737 024001 001170      BIC    #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
5338 031350 013737 001170 001164      MOV    STMP2,STMP0   ;COPY IT INTO 'STMP0'
5339 031356 042737 100100 001164      ZIC    #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5340 031364 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
5341 031372 016037 000012 001172      MOV    RMS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5342 031400 042737 024001 001172      BIC    #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
5343 031406 013737 001172 001166      MOV    STMP3,STMP1  ;COPY IT INTO 'STMP1'
5344 031414 042737 100100 001166      BIC    #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5345 031422 023737 001164 001166      CMP    STMP0,STMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5346 031430 001006      BNE    64$          ;BR IF NOT
5347 031432 005737 001164      TST   STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5348 031436 001045      BNE    66$          ;BR IF NOT
5349 031440 104046      ERROR  46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5350 031442 000137 031642      JMP    68$          ;BYPASS THE REST OF THE CHECKS
5351 031446 013737 001170 001126 64$:  MOV    STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5352 031454 013737 001226 001240      MOV    PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5353 031462 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
5354 031470 005737 001164      TST   STMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
5355 031474 001414      BEQ    65$          ;BR IF ZERO
5356 031476 013737 001224 001240      MOV    PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5357 031504 013737 001172 001126      MOV    STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5358 031512 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
5359 031520 005737 001166      TST   STMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
5360 031524 001012      BNE    66$          ;BR IF NOT
5361 031526 012737 177777 001254 65$:  MOV    #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
5362 031534 012760 000011 000000      MOV    #11,RMCS1(RO) ;CLEAR THE DRIVE
5363 031542 012760 000013 000000      MOV    #13,RMCS1(RO) ;RELEASE THE DRIVE
5364 031550 104021      ERROR  21          ;TYPE ERROR MESSAGE 21
5365 031552 013737 001170 001126 66$:  MOV    STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
5366 031560 013737 001224 001240      MOV    PORTA,PTNBR  ;CHANGE PORT NUMBER
5367 031566 042737 100000 001126      BIC    #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
5368 031574 023737 001124 001126      CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
5369 031602 001401      BEQ    67$          ;BR IF OK FROM PORT A.
5370 031604 104007      ERROR  7           ;REPORT ERROR
5371 031606 013737 001172 001126 67$:  MOV    STMP3,$BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
5372 031614 013737 001226 001240      MOV    PORTB,PTNBR  ;CHANGE PORT NUMBER
5373 031622 042737 100000 001126      BIC    #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT

```

5374 031630 023737 001124 001126
5375 031636 001401
5376 031640 104007
5377 031642 000240
5378 031644 000004
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399

68\$: CMP \$GDDAT,\$BDDAT ;SEE IF READ OK FROM PORT B.
BEQ 68\$;BR IF OK
ERROR 7 ;REPORT ERROR
NOP
SCOPE ;LOOP ?

;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 RMDS1.
;B. WRITE 0'S INTO RMDS1 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.
;*****

5400 031646
5401 031646 005737 001300
5402 031652 001406
5403 031654 100002
5404 031656 000137 002676
5405 031662 012737 177777 001300
5406 031670 112737 000024 001102
5407 031676 012737 031720 001106
5408 031704 012737 031720 001110
5409 031712 012737 000031 001176
5410 031720 012706 001100
5411
5412
5413
5414 031724 113760 001224 000010
5415 031732 005060 000012
5416 031736 012760 000011 000000
5417 031744 012760 000013 000000
5418 031752 113760 001226 000010
5419 031760 005060 000012
5420 031764 012760 000011 000000
5421 031772 012760 000013 000000
5422
5423
5424
5425
5426
5427 032000 113760 001226 000010
5428 032006 013737 001226 001242
5429 032014 005060 000012

†ST24: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
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\$: MOVB #24,\$STNM ;TEST NUMBER
MOV #TEST24,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST24,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,\$TIMES ;DO 25. ITERATIONS
TEST24: MOV #STACK,\$P ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS

MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
CLR RMDS1(RO) ;SEIZE THE DRIVE
MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(RO) ;RELEASE THE DRIVE

;;*****

;SEIZE THE DRIVE THROUGH PORT B

MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDS1(RO) ;WRITE RMDS1

```

5430 032020 013737 001224 001244      MOV     PORTA,OPPRT      ;'OPPOSITE' PORT ADDRESS
5431 032026 113760 001224 000010      MOV     PORTA,RMCS2(RO) ;SELECT PORT A
5432 032034 013737 001224 001240      MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5433
5434 ;*****
5435 ;SET PORT REQUEST
5436
5437 032042 005060 000012      CLR     RMDS1(RO)       ;SET PORT REQUEST FOR PORT A
5438
5439 ;*****
5440 ;RELEASE THROUGH PORT B. DRIVE SHOULD SWITCH TO PORT A.
5441
5442
5443 ;RELEASE THE DRIVE FROM PORT B
5444
5445 032046 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B
5446 032054 013737 001226 001240      MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5447 032062 012760 000013 000000      MOV     #13,RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT B
5448
5449 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5450
5451 032070 005037 001254      CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5452 032074 012737 111700 001124      MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5453 032102 012737 000012 001122      MOV     #RMDS1,$BDDADR  ;REGISTER ADDRESS INCREMENT
5454 032110 060037 001122      ADD     RO,$BDDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
5455 032114 113760 001224 000010      MOV     PORTA,RMCS2(RO) ;SELECT PORT A
5456 032122 013737 001224 001240      MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5457 032130 016037 000012 001164      MOV     RMDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
5458 032136 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B
5459 032144 013737 001226 001240      MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5460 032152 016037 000012 001126      MOV     RMDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
5461 032160 001404      BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
5462 032162 005737 001164      TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
5463 032166 001401      BEQ     66$             ;BR IF ZERO
5464 032170 104031      ERROR   31             ;REPORT DRIVE IN NEUTRAL
5465 032172 013737 001164 001126 66$:  MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
5466 032200 013737 001224 001240      MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5467 032206 023737 001124 001126      CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5468 032214 001401      BEQ     67$             ;BR IF OK
5469 032216 104027      ERROR   27             ;REPORT REGISTER ERROR
5470 032220 000240 67$:  NOP
5471 032222 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B
5472 032230 013737 001226 001240      MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5473 032236 005037 001250      CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5474 032242 016037 000012 001126      MOV     RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
5475 032250 012737 000012 001122      MOV     #RMDS1,$BDDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5476 032256 060037 001122      ADD     RO,$BDDADR      ;ADD RH11 BASE ADDRESS
5477 032262 005037 001124      CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5478 032266 013737 001126 001164      MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5479 032274 042737 077777 001164      BIC     #1CATA,$TMP0    ;SAVE SPECIFIED BITS
5480 032302 023737 001124 001164      CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5481 032310 001414      BEQ     68$             ;BR IF OK
5482 032312 013737 001126 001174      MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5483 032320 042737 100000 001174      BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5484 032326 053737 001174 001124      BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
5485 032334 104016      ERROR   16             ;TYPE MESSAGE 16
    
```


M08

MO-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 103
 T24 SET PORT 'A' REQUEST TEST

SEQ 0105

```

5486 032336 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
5487 032342 000240          68$:    NOP
5488 032344 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
5489 032352 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5490 032360 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
5491 032364 016037 000012 001126      MOV      RMDS1(RO),SBD0AT ;GET CONTENTS OF RMDS1
5492 032372 012737 000012 001122      MOV      #RMDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5493 032400 060037 001122          ADD      RO,SBDADR      ;ADD RHI1 BASE ADDRESS
5494 032404 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
5495 032412 013737 001126 001164      MOV      SBD0AT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5496 032420 042737 077777 001164      BIC      #1CATA,$TMP0 ;SAVE SPECIFIED BITS
5497 032426 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
5498 032434 001414          BEQ      70$           ;BR IF OK
5499 032436 013737 001126 001174      MOV      SBD0AT,$TMP4 ;COPY 'BAD DATA'
5500 032444 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
5501 032452 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5502 032460 104016          ERROR   16           ;TYPE MESSAGE 16
5503 032462 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
5504 032466 000240          70$:    NOP
5505
5506 ;*****
5507
5508 ;RELEASE THE DRIVE FROM PORT A
5509
5510 032470 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
5511 032476 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5512 032504 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5513
5514 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5515
5516 032512 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
5517 032516 012737 000012 001122      MOV      #RMDS1,SBDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5518 032524 060037 001122          ADD      RO,SBDADR      ;ADD THE I/O BASE ADDRESS
5519 032530 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5520 032536 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A.
5521 032544 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5522 032552 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
5523 032560 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5524 032566 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5525 032574 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B.
5526 032602 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5527 032610 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
5528 032616 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5529 032624 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5530 032632 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5531 032640 001006          BNE     72$           ;BR IF NOT
5532 032642 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5533 032646 001045          BNC     74$           ;BR IF NOT
5534 032650 104046          ERROR   46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5535 032652 000137 033036          JMP     76$           ;BYPASS THE REST OF THE CHECKS
5536 032656 013737 001170 001126 72$:    MOV      $TMP2,$SBD0AT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5537 032664 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5538 032672 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B.
5539 032700 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
5540 032704 001414          BEQ     73$           ;BR IF ZERO
5541 032706 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

T24 SET PORT 'A' REQUEST TEST

```

5542 032714 013737 001172 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5543 032722 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
5544 032730 005737 001166 IST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5545 032734 001012 BRE 74$ ;BR IF NOT
5546 032736 012737 177777 001254 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5547 032744 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
5548 032752 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5549 032760 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5550 032762 013737 001170 001126 74$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5551 032770 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5552 032776 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
5553 033004 001401 BEQ 75$ ;BR IF OK FROM PORT A.
5554 033006 104007 ERROR 7 ;REPORT ERROR
5555 033010 013737 001172 001126 75$: MOV STMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5556 033016 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5557 033024 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5558 033032 001401 BEQ 76$ ;BR IF OK
5559 033034 104007 ERROR 7 ;REPORT ERROR
5560 033036 000240 76$: NOP
5561 033040 000004 1$: SCOPE ;LOOP ?

```

```

*****
*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 RMDS1.
*
* B. WRITE 0'S INTO RMDS1 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.
*
*****

```

```

5581
5582 033042
5583 033042 005737 001300 TST25: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5584 033046 001406 BEQ 2$ ;BR IF NOT
5585 033050 100002 BPL 1$ ;BR IF JUST ENTERED TEST
5586 033052 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5587 033056 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
5588 033064 112737 000025 001102 2$: MOVB #25,$STSTNM ;TEST NUMBER
5589 033072 012737 033114 001106 MOV #TEST25,$LPADR ;LOAD LOOP ON TEST ADDRESS
5590 033100 012737 033114 001110 .JV #TEST25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5591 033106 012737 000031 001176 MOV #25,$TIMES ;DO 25. ITERATIONS
5592 033114 012706 001100 TEST25: MOV #STACK,SP ;LOAD THE STACK POINTER
5593
5594 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5595
5596 033120 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
5597 033126 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE

```

```

5598 033132 012760 000011 000000      MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5599 033140 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5600 033146 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT #B
5601 033154 005060 000012          CLR      RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
5602 033160 012760 000011 000000      MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5603 033166 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5604
5605 ;*****
5606
5607 ;SEIZE THE DRIVE THROUGH PORT A
5608
5609 033174 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A
5610 033202 013737 001224 001242      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
5611 033210 005060 000012          CLR      RMDS1(RO) ;WRITE RMDS!
5612 033214 013737 001226 001244      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
5613 033222 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B
5614 033230 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5615
5616 ;*****
5617 ;SET PORT REQUEST
5618
5619 033236 005060 000012          CLR      RMDS1(RO) ;SET PORT REQUEST FOR PORT B
5620
5621 ;*****
5622 ;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.
5623
5624
5625 ;RELEASE THE DRIVE FROM PORT A
5626
5627 033242 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A
5628 033250 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5629 033256 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5630
5631 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5632
5633 033264 005037 001254          CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5634 033270 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CO. ;TANT
5635 033276 012737 000012 001122      MOV      #RMDS1,$BDDADR ;REGISTER ADDRESS INCREMENT
5636 033304 060037 001122          ADD      RO,$BDDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
5637 033310 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B
5638 033316 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5639 033324 016037 000012 001164      MOV      RMDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT B
5640 033332 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A
5641 033340 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5642 033346 016037 000012 001126      MOV      RMDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
5643 033354 001404          BEQ      66$ ;BR IF STATUS FROM PORT A ZERO
5644 033356 005737 001164          TST      $TMPD ;IS STATUS FROM PORT B ZERO ?
5645 033362 001401          BEQ      66$ ;BR IF ZERO
5646 033364 104031          ERROR   31 ;REPORT DRIVE IN NEUTRAL
5647 033366 013737 001164 001126 66$: MOV      $TMPD,$BDDAT ;CHECK STATUS FROM PORT B
5648 033374 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5649 033402 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
5650 033410 001401          BEQ      67$ ;BR IF OK
5651 033412 104027          ERROR   27 ;REPORT REGISTER ERROR
5652 033414 000240          NOP
5653 033416 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A

```

```

5654 033424 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5655 033432 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5656 033436 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
5657 033444 012737 000012 001122 MOV #RMD51,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5658 033452 060037 001122 ADD RO,SBDADR ;ADD RHI1 BASE ADDRESS
5659 033456 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5660 033462 013737 001126 001164 MOV SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5661 033470 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
5662 033476 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5663 033504 001414 BEQ 68$ ;BR IF OK
5664 033506 013737 001126 001174 MOV SBDAT,$TMP4 ;COPY 'BAD DATA'
5665 033514 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
5666 033522 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5667 033530 104016 ERROR 16 ;TYPE MESSAGE 16
5668 033532 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5669 033536 000240 68$: NOP
5670 033540 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
5671 033546 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5672 033554 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5673 033560 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
5674 033566 012737 000012 001122 MOV #RMD51,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5675 033574 060037 001122 ADD RO,SBDADR ;ADD RHI1 BASE ADDRESS
5676 033600 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
5677 033606 013737 001126 001164 MOV SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5678 033614 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
5679 033622 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5680 033630 001414 BEQ 70$ ;BR IF OK
5681 033632 013737 001126 001174 MOV SBDAT,$TMP4 ;COPY 'BAD DATA'
5682 033640 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
5683 033646 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5684 033654 104016 ERROR 16 ;TYPE MESSAGE 16
5685 033656 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5686 033662 000240 70$: NOP
5687
5688 ;*****
5689
5690 ;RELEASE THE DRIVE FROM PORT B
5691
5692 033664 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
5693 033672 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5694 033700 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5695
5696 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5697
5698 033706 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5699 033712 012737 000012 001122 MOV #RMD51,SBDADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
5700 033720 060037 001122 ADD RO,SBDADR ;ADD THE I/O BASE ADDRESS
5701 033724 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5702 033732 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
5703 033740 016037 000012 001170 MOV RMD51(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5704 033746 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
5705 033754 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5706 033762 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5707 033770 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
5708 033776 016037 000012 001172 MOV RMD51(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5709 034004 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES

```

```

5710 034012 013737 001172 001166      MOV      STMP3,STMP1      ;COPY IT INTO 'STMP1'
5711 034020 042737 100100 001166      BIC      #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5712 034026 023737 001164 001166      CMP      STMP0,STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5713 034034 001006                BNE      72$           ;BR IF NOT
5714 034036 005737 001164                TST      STMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5715 034042 001045                BNE      74$           ;BR IF NOT
5716 034044 104046      ERROR    46         ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5717 034046 000137 034232      JMP      76$           ;BYPASS THE REST OF THE CHECKS
5718 034052 013737 001170 001126 72$:      MOV      STMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5719 034060 013737 001226 001240      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5720 034066 113760 001226 000010      MOVVB   PORTB,RMCS2(RO);SELECT PORT B.
5721 034074 005737 001164                TST      STMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
5722 034100 001414                BEQ      73$           ;BR IF ZERO
5723 034102 013737 001224 001240      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5724 034110 013737 001172 001126      MOV      STMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
5725 034116 113760 001224 000010      MOVVB   PORTA,RMCS2(RO);SELECT PORT A.
5726 034124 005737 001166                TST      STMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
5727 034130 001012                BNE      74$           ;BR IF NOT
5728 034132 012737 177777 001254 73$:      MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
5729 034140 012760 000011 000000      MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
5730 034146 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5731 034154 104026      ERROR    26         ;TYPE ERROR MESSAGE 26
5732 034156 013737 001170 001126 74$:      MOV      STMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5733 034164 013737 001224 001240      MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
5734 034172 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
5735 034200 001401                BEQ      75$           ;BR IF OK FROM PORT A.
5736 034202 104007      ERROR    7         ;REPORT ERROR
5737 034204 013737 001172 001126 75$:      MOV      STMP3,$BDDAT   ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5738 034212 013737 001226 001240      MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
5739 034220 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5740 034226 001401                BEQ      76$           ;BR IF OK
5741 034230 104007      ERROR    7         ;REPORT ERROR
5742 034232 000240 76$:      NOP
5743 034234 000004 1$:      SCOPE                ;LOOP ?

```

5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765

```

*****
*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 RMDS1.
*
* 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.
*****

```

```

5766 034236          TST26:
5767 034236 005737 001300      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
5768 034242 001406          BEQ      25          ;BR IF NOT
5769 034244 100002          BPL      15          ;BR IF JUST ENTERED TEST
5770 034246 000137 002676      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
5771 034252 012737 177777 001300 15:  MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
5772 034260 112737 000026 001102 25:  MOVVB   #26,$STSTNM ;TEST NUMBER
5773 034266 012737 034310 001106      MOV      #TEST26,$LPADR ;LOAD LOOP ON TEST ADDRESS
5774 034274 012737 034310 001110      MOV      #TEST26,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5775 034302 012737 000031 001176      MOV      #25,$TIMES   ;DO 25. ITERATIONS
5776 034310 012706 001100      TEST26: MOV     #STACK,SP ;LOAD THE STACK POINTER
5777 ;*****
5778 ;
5779 ;SET ATTENTION BITS FOR BOTH PORTS
5780
5781 034314 113760 001224 000010      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT 64$
5782 034322 005760 000012      66$:  TST      RMD$1(RO)     ;MAKE SURE DRIVE AVAILABLE
5783 034326 001775          BEQ      66$
5784 034330 012760 177777 000014      MOV      #-1, RMER1(RO) ;FORCE ERRORS
5785 034336 005060 000014      CLR      RMER1(RO)     ;CLEAR THE ERRORS
5786 034342 013760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
5787 034350 005760 000012      64$:  TST      RMD$1(RO)     ;WAIT FOR DRIVE TO TIMEOUT
5788 034354 001775          BEQ      64$          ;BR IF DRIVE HASN'T TIMED OUT
5789 034356 012760 177777 000014      MOV      #-1, RMER1(RO) ;FORCE ERRORS ON PORT 65$
5790 034364 005060 000014      CLR      RMER1(RO)     ;CLEAR THE ERRORS
5791 034370 113760 001224 000010      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
5792 034376 005760 000012      65$:  TST      RMD$1(RO)     ;WAIT FOR DRIVE TO TIMEOUT
5793 034402 001775          BEQ      65$          ;BR IF DRIVE HASN'T TIMED OUT
5794
5795 ;*****
5796 ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
5797
5798 034404 113760 001224 000010      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
5799 034412 013737 001224 001240      MOV      PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5800 034420 005037 001250          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
5801 034424 016037 000012 001126      MOV      RMD$1(RO), $BDDAT ;GET CONTENTS OF RMD$1
5802 034432 012737 000012 001122      MOV      #RMD$1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5803 034440 060037 001122          ADD      RO, $BDAOR    ;ADD RH11 BASE ADDRESS
5804 034444 012737 100000 001124      MOV      #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
5805 034452 013737 001126 001164      MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5806 034460 042737 077777 001164      BIC      #+CATA, $TMP0 ;SAVE SPECIFIED BITS
5807 034466 023737 001124 001164      CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
5808 034474 001414          BEQ      67$          ;BR IF OK
5809 034476 013737 001126 001174      MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
5810 034504 042737 100000 001174      BIC      #ATA, $TMP4   ;CLEAR THE MASKED BITS
5811 034512 053737 001174 001124      BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5812 034520 104010          ERROR  10          ;REPORT THE ERROR
5813 034522 005137 001250          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
5814 034526 000240          67$:  NOP
5815 034530 005737 001250          TST      CKERR        ;WAS ATTN BIT FOR PORT A SET ?
5816 034534 001402          BEQ      .+6         ;BR IF IT WAS
5817 034536 000137 035730          JMP      15          ;BYPASS REST OF TEST IF NOT
5818 034542 113760 001226 000010      MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
5819 034550 013737 001226 001240      MOV      PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5820 034556 005037 001250          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
5821 034562 016037 000012 001126      MOV      RMD$1(RO), $BDDAT ;GET CONTENTS OF RMD$1

```

F09

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T26

MACY11 30(1046) 01-AUG-77 11:02 PAGE 109
TEST RESET ATTENTION 'A' BY DRIVE CLEAR

SEQ 0111

```

5822 034570 012737 000012 001122      MOV      #RMS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5823 034576 060037 001122      ADD      R0,$B0ADR   ;ADD R011 BASE ADDRESS
5824 034602 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
5825 034610 013737 001126 001164      MOV      $B0DAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5826 034616 042737 077777 001164      BIC      #ICATA,$TMP0 ;SAVE SPECIFIED BITS
5827 034624 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
5828 034632 001414      BEQ      69$         ;BR IF OK
5829 034634 013737 001126 001174      MOV      $B0DAT,$TMP4 ;COPY 'BAD DATA'
5830 034642 042737 100000 001174      BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
5831 034650 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5832 034656 104010      ERROR   10         ;REPORT THE ERROR
5833 034660 005137 001250      COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
5834 034664 000240      69$:  NOP
5835 034666 005737 001250      TST      CKERR       ;WAS ATTN BIT FOR PORT B SET ?
5836 034672 001402      BEQ      +6         ;BR IF IT WAS
5837 034674 000137 035730      JMP      1$         ;BYPASS REST OF TEST IF NOT

```

;;*****

;SEIZE THE DRIVE THROUGH PORT A

```

5843 034700 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
5844 034706 013737 001224 001242      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
5845 034714 005060 000012      CLR      RMS1(R0)      ;WRITE RMS1
5846 034720 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
5847 034726 013737 001226 001240      MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5848 034734 013737 001226 001244      MOV      PORTB,OPPR   ;'OPPOSITE' PORT ADDRESS
5849 034742 016037 000012 001126      MOV      RMS1(R0),$B0DAT ;SEE IF DRIVE SEIZED BY PORT A
5850 034750 010037 001122      MOV      R0,$B0ADR    ;R011 BASE ADDRESS
5851 034754 062737 000012 001122      ADD      #RMS1,$B0ADR  ;GENERATE BAD REGISTER ADDRESS
5852 034762 005037 001124      CLR      $GDDAT       ;REGISTER SHOULD BE ZERO
5853 034766 023737 001124 001126      CMP      $GDDAT,$B0DAT ;IS THE REGISTER ZERO
5854 034774 001403      BEQ      71$         ;BR IF IT IS
5855 034776 104004      ERROR   4          ;REPORT THE ERROR
5856 035000 000137 035730      JMP      1$         ;BYPASS REST OF THE SUBTEST

```

71\$:

```

5858 035004 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
5859 035012 013737 001224 001240      MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5860 035020 016037 000012 001126      MOV      RMS1(R0),$B0DAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5861 035026 042737 020001 001126      BIC      #OM!PIP,$B0DAT ;CLEAR DONT CARE BITS
5862 035034 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5863 035042 013737 001124 001166      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5864 035050 005137 001166      COM      $TMP1        ;COMPLEMENT THE EXPECTED STATUS
5865 035054 013737 001126 001164      MOV      $B0DAT,$TMP0 ;SAVE THE ACTUAL STATUS
5866 035062 043737 001166 001164      BIC      $TMP1,$TMP0  ;CLEAR UNWANTED BITS
5867 035070 023737 001124 001164      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5868 035076 001401      BEQ      72$         ;BR IF THEY ARE
5869 035100 104005      ERROR   5          ;REPORT THE ERROR

```

72\$:

```

5870 035102 000240      NOP
5871
5872
5873
5874
5875 035104 012760 000011 000000      MOV      #11,RMCS1(R0) ;DO A DRIVE CLEAR COMMAND

```

;;*****

;ISSUE DRIVE CLEAR COMMAND TO PORT A

;;*****

;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED

5878										
5879										
5880	035112	005037	001250			CLR	CKERR		;CLEAR THE 'CHECK ERROR' INDICATOR	
5881	035116	016037	000012	001126		MOV	RMDS1(RO),SBD0AT		;GET CONTENTS OF RMDS1	
5882	035124	012737	000012	001122		MOV	#RMDS1,SBDADR		;FORM REGISTER ADDRESS OF ERROR MESSAGE	
5883	035132	060037	001122			ADD	RO,SBDADR		;ADD RH11 BASE ADDRESS	
5884	035136	005037	001124			CLR	SGD0AT		;WHAT REGISTER SHOULD BE	
5885	035142	013737	001126	001164		MOV	SBD0AT,\$TMP0		;MOVE REGISTER CONTENTS TO 'STMP0'	
5886	035150	042737	077777	001164		BIC	#ICATA,\$TMP0		;SAVE SPECIFIED BITS	
5887	035156	023737	001124	001164		CMP	SGD0AT,\$TMP0		;COMPARE THE BITS	
5888	035164	001414				BEQ	73\$;BR IF OK	
5889	035166	013737	001126	001174		MOV	SBD0AT,\$TMP4		;COPY 'BAD DATA'	
5890	035174	042737	100000	001174		BIC	#ATA,\$TMP4		;CLEAR THE MASKED BITS	
5891	035202	053737	001174	001124		BIS	\$TMP4,\$GD0AT		; 'OR' WITH GOOD DATA FOR TYPEOUT	
5892	035210	104047				ERROR	47		;TYPE MESSAGE 47	
5893	035212	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR	
5894	035216	000240				73\$:	NOP			
5895										
5896									;*****	
5897										
5898									;RELEASE THE DRIVE FROM PORT A	
5899										
5900	035220	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A	
5901	035226	013737	001224	001240		MOV	PORTA,PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT	
5902	035234	012760	000013	000000		MOV	#13,RMCS1(RO)		;ISSUE RELEASE THROUGH PORT A	
5903										
5904									;VERIFY THAT THE DRIVE IS IN NEUTRAL	
5905										
5906	035242	005037	001254			CLR	RELERR		;CLEAR THE 'RELEASE ERROR' INDICATOR	
5907	035246	012737	000012	001122		MOV	#RMDS1,SBDADR		;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT	
5908	035254	060037	001122			ADD	RO,SBDADR		;ADD THE I/O BASE ADDRESS	
5909	035260	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GD0AT		;COMPARISON CONSTANT	
5910	035266	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A.	
5911	035274	016037	000012	001170		MOV	RMDS1(RO),\$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.	
5912	035302	042737	024001	001170		BIC	#PIP!WRL!OM,\$TMP2		;CLEAR DONT CARES	
5913	035310	013737	001170	001164		MOV	\$TMP2,\$TMP0		;COPY IT INTO 'STMP0'	
5914	035316	042737	100100	001164		BIC	#ATA!VV,\$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY	
5915	035324	113760	001226	000010		MOVB	PORTB,RMCS2(RO)		;SELECT PORT B.	
5916	035332	016037	000012	001172		MOV	RMDS1(RO),\$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.	
5917	035340	042737	024001	001172		BIC	#PIP!WRL!OM,\$TMP3		;CLEAR DONT CARES	
5918	035346	013737	001172	001166		MOV	\$TMP3,\$TMP1		;COPY IT INTO 'STMP1'	
5919	035354	042737	100100	001166		BIC	#ATA!VV,\$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY	
5920	035362	023737	001164	001166		CMP	\$TMP0,\$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?	
5921	035370	001006				BNE	75\$;BR IF NOT	
5922	035372	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?	
5923	035376	001045				BNE	77\$;BR IF NOT	
5924	035400	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED	
5925	035402	000137	035602			JMP	79\$;BYPASS THE REST OF THE CHECKS	
5926	035406	013737	001170	001126	75\$:	MOV	\$TMP2,SBD0AT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE	
5927	035414	013737	001226	001240		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
5928	035422	113760	001226	000010		MOVB	PORTB,RMCS2(RO)		;SELECT PORT B.	
5929	035430	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.	
5930	035434	001414				BEQ	76\$;BR IF ZERO	
5931	035436	013737	001224	001240		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
5932	035444	013737	001172	001126		MOV	\$TMP3,SBD0AT		; 'BAD DATA' FOR ERROR TYPE OUT	
5933	035452	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A.	


```

5934 035460 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
5935 035464 001012          BNE      77$           ;BR IF NOT
5936 035466 012737 177777 001254 76$:  MOV     #-1,RELEP     ;SET 'RELEASE ERROR' INDICATOR
5937 035474 012760 000011 000000  MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
5938 035502 012760 000013 000000  MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
5939 035510 104026          ERROR   26           ;TYPE ERROR MESSAGE 26
5940 035512 013737 001170 001126 77$:  MOV     $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5941 035520 013737 001224 001240  MOV     PORTB,PTNBR   ;CHANGE PORT NUMBER
5942 035526 042737 100000 001126  BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
5943 035534 023737 001124 001126  CMP     $GDDAT,$BDDAT ;ALL BITS OK
5944 035542 001401          BEQ     78$           ;BR IF OK FROM PORT A.
5945 035544 104007          ERROR   7           ;REPORT ERROR
5946 035546 013737 001172 001126 78$:  MOV     $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5947 035548 013737 001226 001240  MOV     PORTB,PTNBR   ;CHANGE PORT NUMBER
5948 035550 042737 100000 001126  BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
5949 035570 023737 001124 001126  CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5950 035576 001401          BEQ     79$           ;BR IF OK
5951 035600 104007          ERROR   7           ;REPORT ERROR
5952 035602 000240          NOP
5953
5954
5955
5956
5957 035604 113760 001226 000010  MOV     PORTB,RMCS2(RO) ;SELECT PORT B
5958 035612 013737 001226 001240  MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5959 035620 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
5960 035624 016037 000012 001126  MOV     RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
5961 035632 012737 000012 001122  MOV     #RMDS1,$BDDADR ;FCR REGISTER ADDRESS OF ERROR MESSAGE
5962 035640 060037 001122          ADD     R0,$BDDADR   ;ADD RHI1 BASE ADDRESS
5963 035644 012737 100000 001124  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
5964 035652 013737 001126 001164  MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO 'TMP0'
5965 035660 042737 077777 001164  BIC     #+DATA,$TMP0 ;SAVE SPECIFIED BITS
5966 035666 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
5967 035674 001414          BEQ     80$           ;BR IF OK
5968 035676 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
5969 035704 042737 100000 001174  BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
5970 035712 053737 001174 001124  BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
5971 035720 104050          ERROR   50          ;TYPE MESSAGE 50
5972 035722 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
5973 035726 000240          NOP
5974 035730          IS:  SCOPE          ;LOOP ?
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989

```

```

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

```

```

;*****
;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 RMDS1.
;
; C. ISSUE A DRIVE CLEAR COMMAND.
;
; D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION
;

```

```

5990 ;* BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
5991 ;* 'A' IS STILL SET.
5992 ;*
5993 ;*****
5994 TST27: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5995 035732 005737 001300 BEQ 25 ;BR IF NOT
5996 035736 001406 BPL 15 ;BR IF JUST ENTERED TEST
5997 035740 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5998 035742 000137 002676 15: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
5999 035746 012737 177777 001300 25: MOV #27, $STNM ;TEST NUMBER
6000 035754 112737 000027 001102 MOV #TEST27, $LPADR ;LOAD LOOP ON TEST ADDRESS
6001 035762 012737 036004 001106 MOV #TEST27, $LPERR ;LOAD LOOP ON ERROR ADDRESS
6002 035770 012737 036004 001110 MOV #25, $TIMES ;DO 25. ITERATIONS
6003 035776 012737 000031 001176 TEST27: MOV #STACK, SP ;LOAD THE STACK POINTER
6004 036004 012706 001100 ;*****
6005 ;
6006 ;SET ATTENTION BITS FOR BOTH PORTS
6007 ;
6008 ;
6009 036010 113760 001224 000010 66$: MOVB PORTA, RMCS2(RO) ;SELECT PORT 64$
6010 036016 005760 000012 TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
6011 036022 001775 BEQ 66$
6012 036024 012760 177777 000014 MOV #-1, RMER1(RO) ;FORCE ERRORS
6013 036032 005060 000014 CLR RMER1(RO) ;CLEAR THE ERRORS
6014 036036 013760 001226 000010 64$: MOV PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
6015 036044 005760 000012 TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6016 036050 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
6017 036052 012760 177777 000014 MOV #-1, RMER1(RO) ;FORCE ERRORS ON PORT 65$
6018 036060 005060 000014 CLR RMER1(RO) ;CLEAR THE ERRORS
6019 036064 113760 001224 000010 65$: MOVB PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
6020 036072 005760 000012 TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6021 036076 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
6022 ;
6023 ;*****
6024 ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
6025 ;
6026 036100 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
6027 036106 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6028 036114 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6029 036120 016037 000012 001126 MOV RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
6030 036126 012737 000012 001122 MOV #RMD51, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6031 036134 060037 001122 ADD RO, $B0ADR ;ADD RHI1 BASE ADDRESS
6032 036140 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6033 036146 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
6034 036154 042737 077777 001164 BIC #+CATA, $TMP0 ;SAVE SPECIFIED BITS
6035 036162 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6036 036170 001414 BEQ 67$ ;BR IF OK
6037 036172 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6038 036200 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
6039 036206 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6040 036214 104010 ERROR 10 ;REPORT THE ERROR
6041 036216 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6042 036222 000240 67$: NOP
6043 036224 005737 001250 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
6044 036230 001402 BEQ .+6 ;BR IF IT WAS
6045 036232 000137 037424 JMP 15 ;BYPASS REST OF TEST IF NOT

```

```

6046 036236 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
6047 036244 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6048 036252 005037 001250                CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6049 036256 016037 000012 001126      MOV   RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
6050 036264 012737 000012 001122      MOV   #RMD51, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6051 036272 060037 001122                ADD   RO, $BDAOR ;ADD RH11 BASE ADDRESS
6052 036276 012737 100000 001124      MOV   #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6053 036304 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6054 036312 042737 077777 001164      BIC   #1CATA, $TMP0 ;SAVE SPECIFIED BITS
6055 036320 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
6056 036326 001414                BEQ   69$ ;BR IF OK
6057 036330 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
6058 036336 042737 100000 001174      BIC   #ATA, $TMP4 ;CLEAR THE MASKED BITS
6059 036344 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6060 036352 104010      ERROR 10 ;REPORT THE ERROR
6061 036354 005137 001250                COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6062 036360 000240      69$: NOP
6063 036362 005737 001250                TST   CKERR ;WAS ATTN BIT FOR PORT A SET ?
6064 036366 001402                BEQ   .+6 ;BR IF IT WAS
6065 036370 000137 037424                JMP   1$ ;BYPASS REST OF TEST IF NOT

```

;;*****

;SEIZE THE DRIVE THROUGH PORT B

```

6071 036374 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
6072 036402 013737 001226 001242      MOV   PORTB, SFIZP1 ;STORE SEIZING PORT'S ADDRESS
6073 036410 005060 000012                CLR   RMD51(RO) ;WRITE RMD51
6074 036414 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
6075 036422 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6076 036430 013737 001224 001244      MOV   PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
6077 036436 016037 000012 001126      MOV   RMD51(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
6078 036444 010037 001122                MOV   RO, $BDAOR ;RH11 BASE ADDRESS
6079 036450 062737 000012 001122      ADD   #RMD51, $BDAOR ;GENERATE BAD REGISTER ADDRESS
6080 036456 005037 001124                CLR   $GDDAT ;REGISTER SHOULD BE ZERO
6081 036462 023737 001124 001126      CMP   $GDDAT, $BDDAT ;IS THE REGISTER ZERO
6082 036470 001403                BEQ   71$ ;BR IF IT IS
6083 036472 104004      ERROR 4 ;REPORT THE ERROR
6084 036474 000137 037424                JMP   1$ ;BYPASS REST OF THE SUBTEST

```

71\$:

```

6085 036500      71$: MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
6086 036500 113760 001226 000010      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6087 036506 013737 001226 001240      MOV   RMD51(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
6088 036514 016037 000012 001126      BIC   #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
6089 036522 042737 020001 001126      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
6090 036530 012737 011700 001124      MOV   $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
6091 036536 013737 001124 001166      COM   $TMP1 ;COMPLEMENT THE EXPECTED STATUS
6092 036544 005137 001166                MOV   $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
6093 036550 013737 001126 001164      BIC   $TMP1, $TMP0 ;CLEAR UNWANTED BITS
6094 036556 043737 001166 001164      CMP   $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
6095 036564 023737 001124 001164      BEQ   72$ ;BR IF THEY ARE
6096 036572 001401      ERROR 5 ;REPORT THE ERROR
6097 036574 104005
6098 036576 000240      72$: NOP

```

72\$:

;;*****
;ISSUE DRIVE CLEAR COMMAND TO PORT B

6100
6101

```

6102
6103 036600 012760 000011 000000      MOV      #11, RMCS1(RO) ;DO A DRIVE CLEAR COMMAND
6104
6105 ;*****
6106 ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
6107
6108 036606 005037 001250      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
6109 036612 016037 000012 001126      MOV      RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
6110 036620 012737 000012 001122      MOV      #RMDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6111 036626 050037 001122      ADD      RO, $BDAOR      ;ADD RH11 BASE ADDRESS
6112 036632 005037 001124      CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
6113 036636 013737 001126 001164      MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
6114 036644 042737 077777 001164      BIC      #1CATA, $TMP0   ;SAVE SPECIFIED BITS
6115 036652 023737 001124 001164      CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
6116 036660 001414          BEQ      73$            ;BR IF OK
6117 036662 013737 001126 001174      MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
6118 036670 042737 100000 001174      BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
6119 036676 053737 001174 001124      BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
6120 036704 104047          ERROR    47           ;TYPE MESSAGE 47
6121 036706 005137 001250      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
6122 036712 000240      73$:  NOP
6123
6124 ;*****
6125 ;
6126 ;RELEASE THE DRIVE FROM PORT B
6127
6128 036714 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
6129 036722 013737 001226 001240      MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6130 036730 012760 000013 000000      MOV      #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
6131
6132 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6133
6134 036736 005037 001254      CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
6135 036742 012737 000012 001122      MOV      #RMDS1, $BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
6136 036750 060037 001122      ADD      RO, $BDAOR      ;ADD THE I/O BASE ADDRESS
6137 036754 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
6138 036762 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A.
6139 036770 016037 000012 001170      MOV      RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6140 036776 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
6141 037004 013737 001170 001164      MOV      $TMP2, $TMP0    ;COPY IT INTO 'TMP0'
6142 037012 042737 100100 001164      BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6143 037020 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
6144 037026 016037 000012 001172      MOV      RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6145 037034 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
6146 037042 013737 001172 001166      MOV      $TMP3, $TMP1    ;COPY IT INTO 'TMP1'
6147 037050 042737 100100 001166      BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6148 037056 023737 001164 001166      CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6149 037064 001006          BNE      75$          ;BR IF NOT
6150 037066 005737 001164          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6151 037072 001045          BNE      77$          ;BR IF NOT
6152 037074 104046          ERROR    46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6153 037076 000137 037276          JMP      79$          ;BYPASS THE REST OF THE CHECKS
6154 037102 013737 001170 001126 75$:  MOV      $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6155 037110 013737 001226 001240      MOV      PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6156 037116 113760 001226 000010      MOVE     PORTB, RMCS2(RO) ;SELECT PORT B.
6157 037124 005737 001164          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.

```

```

6158 037130 001414          BEQ      76$          ;BR IF ZERO
6159 037132 013737 001224 001240    MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6160 037140 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6161 037146 113760 001224 000010    MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.
6162 037154 005737 001166          TST     $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
6163 037160 001012          BNE     77$          ;BR IF NOT
6164 037162 012737 177777 001254 76$:  MOV     #-1,RELERR  ;SET 'RELEASE ERROR' INDICATOR
6165 037170 012760 000011 000000    MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
6166 037176 012760 000013 000000    MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
6167 037204 104026          ERROR   26          ;TYPE ERROR MESSAGE 26
6168 037206 013737 001170 001126 77$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD51 READ
6169 037214 013737 001224 001240    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
6170 037222 042737 100000 001126    BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
6171 037230 023737 001124 001126    CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
6172 037236 001401          BEQ     78$          ;BR IF OK FROM PORT A.
6173 037240 104007          ERROR   7           ;REPORT ERROR
6174 037242 013737 001172 001126 78$:  MOV     $TMP3,$BDDAT ;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
6175 037250 013737 001226 001240    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
6176 037256 042737 100000 001126    BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
6177 037264 023737 001124 001126    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6178 037272 001401          BEQ     79$          ;BR IF OK
6179 037274 104007          ERROR   7           ;REPORT ERROR
6180 037276 000240 79$:  NOP

```

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

```

6185 037300 113760 001224 000010    MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
6186 037306 013737 001224 001240    MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6187 037314 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
6188 037320 016037 000012 001126    MOV     RMD51(RO),$BDDAT ;GET CONTENTS OF RMD51
6189 037326 012737 000012 001122    MOV     #RMD51,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6190 037334 060037 001122          ADD     RO,$BDDADR  ;ADD RMD51 BASE ADDRESS
6191 037340 012737 100000 001124    MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
6192 037346 013737 001126 001164    MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6193 037354 042737 077777 001164    BIC     #ICATA,$TMP0 ;SAVE SPECIFIED BITS
6194 037362 023737 001124 001164    CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
6195 037370 001414          BEQ     80$          ;BR IF OK
6196 037372 013737 001126 001174    MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
6197 037400 042737 100000 001174    BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
6198 037406 053737 001174 001124    BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6199 037414 104050          ERROR   50          ;TYPE MESSAGE 50
6200 037416 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
6201 037422 000240 80$:  NOP
6202 037424 000004 1$:  SCOPE          ;LOOP ?

```

;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.
*

6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213

T30 RESET ATTENTION 'A' BY GO TEST

```

6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225 037426
6226 037426 005737 001300
6227 037432 001406
6228 037434 100002
6229 037436 000137 002676
6230 037442 012737 177777 001300
6231 037450 112737 000030 001102
6232 037456 012737 037500 001106
6233 037464 012737 037500 001110
6234 037472 012737 000031 001176
6235 037500 012706 001100
6236
6237
6238
6239
6240
6241 037504 113760 001224 000010
6242 037512 005760 000012
6243 037516 001775
6244 037520 012760 177777 000014
6245 037526 005060 000014
6246 037532 013760 001226 000010
6247 037540 005760 000012
6248 037544 001775
6249 037546 012760 177777 000014
6250 037554 005060 000014
6251 037560 113760 001224 000010
6252 037566 005760 000012
6253 037572 001775
6254
6255
6256
6257
6258 037574 113760 001224 000010
6259 037602 013737 001224 001240
6260 037610 005037 001250
6261 037614 016037 000012 001126
6262 037622 012737 000012 001122
6263 037630 060037 001122
6264 037634 012737 100000 001124
6265 037642 013737 001126 001164
6266 037650 042737 077777 001164
6267 037656 023737 001124 001164
6268 037664 001414
6269 037666 013737 001126 001174

```

```

;*
;* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S
;* INTO RMD5.
;*
;* C. ISSUE A NOP COMMAND.
;*
;* D. RELEASE THE FRIVE THROUGH PORT 'A'. VERIFY THAT THE
;* ATTENTION BIT FOR PORT 'A' IS RESET, AND THE
;* ATTENTION BIT FOR PORT 'B' IS STIL SET.
;*****
†ST30:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #30,$STNM ;TEST NUMBER
MOV #TEST30,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST30,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25. ITERATIONS
TEST30: MOV #STACK,SP ;LOAD THE STACK POINTER
;*****
;SET ATTENTION BITS FOR BOTH PORTS
66$: MOVB PORTA,RMCS2(RO) ;SELECT PORT 64$
TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
BEQ 66$
MOV #-1,RMER1(RO) ;FORCE ERRORS
CLR RMER1(RO) ;CLEAR THE ERRORS
64$: MOV PORTB,RMCS2(RO) ;SELECT THE OTHER PORT
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RMER1(RO) ;FORCE ERRORS ON PORT 65$
CLR RMER1(RO) ;CLEAR THE ERRORS
65$: MOVB PORTA,RMCS2(RO) ;SELECT PORT "64$" AGAIN
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA,RMCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
MOV #RMD51,$SBOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$SBOADR ;ADD RHI1 BASE ADDRESS
MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC #CATA,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 67$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'

```

```

6270 037674 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
6271 037702 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
6272 037710 104010 001174 001124      ERROR   10             ;REPORT THE ERROR
6273 037712 005137 001250              COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
6274 037716 000240 001250              675:  NOP              ;
6275 037720 005737 001250              TST      CKERR          ;WAS ATTENTION SET FOR A??
6276 037724 001402 001250              BEQ      .+6            ;YES!!
6277 037726 000137 041120              JMP      1$            ;NO - BYPASS REST OF TEST
6278 037732 113760 001226 000010      MOVb    PORTB,RMCS2(RO) ;SELECT PORT B
6279 037740 013737 001226 001240      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6280 037746 005037 001250              CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
6281 037752 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
6282 037760 012737 000012 001122      MOV      #RMD51,$B0ADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6283 037766 060037 001122              ADD      RO,$B0ADR     ;ADD RHI1 BASE ADDRESS
6284 037772 012737 100000 001124      MOV      #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
6285 040000 013737 001126 001164      MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO 'TMP0'
6286 040006 042737 077777 001164      BIC      #+CATA,$TMP0  ;SAVE SPECIFIED BITS
6287 040014 023737 001124 001164      CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
6288 040022 001414 001124 001164      BEQ      695          ;BR IF OK
6289 040024 013737 001126 001174      MOV      $BDDAT,$TMP4  ;COPY 'BAD DATA'
6290 040032 042737 100000 001174      BIC      #ATA,$TMP4    ;CLEAR THE MASKED BITS
6291 040040 053737 001174 001124      BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
6292 040046 104010 001174 001124      ERROR   10             ;REPORT THE ERROR
6293 040050 005137 001250              COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
6294 040054 000240 001250              695:  NOP              ;
6295 040056 005737 001250              TST      CKERR          ;WAS ATTENTION SET FOR B??
6296 040062 001402 001250              BEQ      .+6            ;YES!!
6297 040064 000137 041120              JMP      1$            ;NO - BYPASS REST OF TEST

```

;*****

;SEIZE THE DRIVE THROUGH PORT A

```

6303 040070 113760 001224 000010      MOVb    PORTA,RMCS2(RO) ;SELECT PORT A
6304 040076 013737 001224 001242      MOV      PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
6305 040104 005060 000012 001242      CLR      RMD51(RO)     ;WRITE RMD51
6306 040110 113760 001226 000010      MOVb    PORTB,RMCS2(RO) ;SELECT PORT B
6307 040116 013737 001226 001240      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6308 040124 013737 001226 001244      MOV      PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
6309 040132 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF DR'VE SEIZED BY PORT A
6310 040140 010037 001122              MOV      RO,$B0ADR     ;RHI1 BASE ADDRESS
6311 040144 062737 000012 001122      ADD      #RMD51,$B0ADR  ;GENERATE BAD REGISTER ADDRESS
6312 040152 005037 001124              CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
6313 040156 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
6314 040164 001403 001124 001126      BEQ      715          ;BR IF IT IS
6315 040166 104004 001124 001126      ERROR   4             ;REPORT THE ERROR
6316 040170 000137 041120              JMP      1$            ;BYPASS REST OF THE SUBTEST
6317 040174 001124 001126              715:  MOVb    PORTA,RMCS2(RO) ;SELECT PORT A
6318 040174 113760 001224 000010      MOVb    PORTA,RMCS2(RO) ;SELECT PORT A
6319 040202 013737 001224 001240      MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6320 040210 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
6321 040216 042737 020001 001126      BIC      #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
6322 040224 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
6323 040232 013737 001124 001166      MOV      $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
6324 040240 005137 001166              COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
6325 040244 013737 001126 001164      MOV      $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS

```

B10

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T30

MACY11 30(1046) 01-AUG-77 11:02 PAGE 118
RESET ATTENTION 'A' BY GO TEST

SEQ 0120

```

6326 040252 043737 001166 001164      BIC      $TMP1,$TMP0      ;CLEAR UNWANTED BITS
6327 040260 023737 001124 001164      CMP      $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
6328 040266 001401                      BEQ      725              ;BR IF THEY ARE
6329 040270 104005                      ERROR 5                    ;REPORT THE ERROR
6330 040272 000240      725:    NOP
6331
6332      ;*****
6333
6334      ;ISSUE NOP COMMAND TO PORT A
6335
6336 040274 012760 000001 000000      MOV      #1,RMCS1(RO)
6337
6338      ;*****
6339      ;VERIFY THAT ATTENTION FOR PORT A CLEARED
6340
6341 040302 005037 001250                      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
6342 040306 016037 000012 001126      MOV      RMDS1(RO),SBOADR ;GET CONTENTS OF RMDS1
6343 040314 012737 000012 001122      MOV      #RMDS1,SBOADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6344 040322 060037 001122                      ADD      RO,SBOADR       ;ADD RHI1 BASE ADDRESS
6345 040326 005037 001124                      CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
6346 040332 013737 001126 001164      MOV      $SDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO 'TMP0'
6347 040340 042737 077777 001164      BIC      #1,CATA,$TMP0   ;SAVE SPECIFIED BITS
6348 040346 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
6349 040354 001414                      BEQ      735              ;BR IF OK
6350 040356 013737 001126 001174      MOV      $SDDAT,$TMP4    ;COPY 'BAD DATA'
6351 040364 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
6352 040372 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6353 040400 104061                      ERROR 61                  ;TYPE MESSAGE 61
6354 040402 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
6355 040406 000240      735:    NOP
6356
6357      ;*****
6358
6359      ;RELEASE THE DRIVE FROM PORT A
6360
6361 040410 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
6362 040416 013737 001224 001240      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6363 040424 012760 000013 000000      MOV      #13,RMCS1(RO)  ;ISSUE RELEASE THROUGH PORT A
6364
6365      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6366
6367 040432 005037 001254                      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
6368 040436 012737 000012 001122      MOV      #RMDS1,SBOADR   ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
6369 040444 060037 001122                      ADD      RO,SBOADR       ;ADD THE I/O BASE ADDRESS
6370 040450 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6371 040456 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
6372 040464 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6373 040472 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
6374 040500 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO 'TMP0'
6375 040506 042737 100100 001164      BIC      #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6376 040514 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
6377 040522 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6378 040530 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
6379 040536 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO 'TMP1'
6380 040544 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6381 040552 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```



```

6382 040560 001006 BNE 75$ ;BR IF NOT
6383 040562 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6384 040566 001045 BNE 77$ ;BR IF NOT
6385 040570 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6386 040572 000137 040772 JMP 79$ ;BYPASS THE REST OF THE CHECKS
6387 040576 013737 001170 001126 75$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6388 040604 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6389 040612 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
6390 040620 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6391 040624 001414 BEQ 76$ ;BR IF ZERO
6392 040626 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6393 040634 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6394 040642 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
6395 040650 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6396 040654 001012 BNE 77$ ;BR IF NOT
6397 040656 012737 177777 001254 76$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6398 040664 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
6399 040672 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
6400 040700 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6401 040702 013737 001170 001126 77$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD51 READ
6402 040710 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6403 040716 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6404 040724 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
6405 040732 001401 BEQ 78$ ;BR IF OK FROM PORT A.
6406 040734 104007 ERROR 7 ;REPORT ERROR
6407 040736 013737 001172 001126 78$: MOV $TMP3,$BDDAT ;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
6408 040744 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6409 040752 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6410 040760 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6411 040766 001401 BEQ 79$ ;BR IF OK
6412 040770 104007 ERROR 7 ;REPORT ERROR
6413 040772 000240 79$: NOP
6414
6415 ;*****
6416
6417 ;VERIFY THAT ATTENTION FOR PORT B IS STILL SET
6418
6419 040774 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
6420 041002 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6421 041010 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6422 041014 016037 000012 001126 MOV RMD51(RO),$BDDAT ;GET CONTENTS OF RMD51
6423 041022 012737 000012 001122 MOV #RMD51,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6424 041030 060037 001122 ADD RO,$BDDADR ;ADD R#11 BASE ADDRESS
6425 041034 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6426 041042 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6427 041050 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
6428 041056 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6429 041064 001414 BEQ 80$ ;BR IF OK
6430 041066 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6431 041074 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6432 041102 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6433 041110 104062 ERROR 62 ;TYPE MESSAGE 62
6434 041112 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6435 041116 000240 80$: NOP
6436 041120 000004 1$: SCOPE
6437

```

6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493

041122
041122 005737 001300
041126 001406
041130 100002
041132 000137 002676
041136 012737 177777 001300
041144 112737 000031 001102
041152 012737 041174 001106
041160 012737 041174 001110
041166 012737 000031 001176
041174 012706 001100

041200 113760 001224 000010
041206 005760 000012
041212 001775
041214 012760 177777 000014
041222 005060 000014
041226 013760 001226 000010
041234 005760 000012
041240 001775
041242 012760 177777 000014
041250 005060 000014
041254 113760 001224 000010
041262 005760 000012
041266 001775

041270 113760 001226 000010
041276 013737 001226 001240
041304 005037 001250

```
*****
*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 RMD5.
*
* C.  ISSUE A NOP COMMAND.
*
* D.  RELEASE THE FRIVE THROUGH PORT 'B'. VERIFY THAT THE
*     ATTENTION BIT FOR PORT 'B' IS RESET, AND THE
*     ATTENTION BIT FOR PORT 'A' IS STIL SET.
*****
```

```
*****
*ST31:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      25          ;BR IF NOT
      BPL      15          ;BR IF JUST ENTERED TEST
      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
15:   MOV      #-1, KYBCTL ;SET SINGLE TEST INDICATOR
25:   MOV      #31, $STNM  ;TEST NUMBER
      MOV      #TEST31, $LPAOR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST31, $LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #25, $TIMES  ;DO 25. ITERATIONS
TEST31: MOV     #STACK, SP ;LOAD THE STACK POINTER
*****
```

```
*****
;SET ATTENTION BITS FOR BOTH PORTS
      MOV      PORTA, RMCS2(RO) ;SELECT PORT 64$
66$:  TST      RMD51(RO)      ;MAKE SURE DRIVE AVAILABLE
      BEQ      66$
      MOV      #-1, RMER1(RO) ;FORCE ERRORS
      CLR      RMER1(RO)      ;CLEAR THE ERRORS
      MOV      PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
64$:  TST      RMD51(RO)      ;WAIT FOR DRIVE TO TIMEOUT
      BEQ      64$           ;BR IF DRIVE HASN'T TIMED OUT
      MOV      #-1, RMER1(RO) ;FORCE ERRORS ON PORT 65$
      CLR      RMER1(RO)      ;CLEAR THE ERRORS
      MOV      PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
65$:  TST      RMD51(RO)      ;WAIT FOR DRIVE TO TIMEOUT
      BEQ      65$           ;BR IF DRIVE HASN'T TIMED OUT
*****
```

```
*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
*****
```

E10

MO-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T31

MACY11 30(1046) 01-AUG-77 11:02 PAGE 121
RESET ATTENTION 'B' BY GO TEST

SEQ 0123

```

6494 041310 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET-CONTENTS OF RMDS1
6495 041316 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6496 041324 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
6497 041330 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6498 041336 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6499 041344 042737 077777 001164 BIC #ICATA, $TMP0 ;SAVE SPECIFIED BITS
6500 041352 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6501 041360 001414 BEQ 67$ ;BR IF OK
6502 041362 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6503 041370 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
6504 041376 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6505 041404 104010 ERROR 10 ;REPORT THE ERROR
6506 041406 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6507 041412 000240 67$: NOP
6508 041414 005737 001250 TST CKERR ;WAS ATTENTION SET FOR B??
6509 041420 001402 BEQ .+6 ;YES!!
6510 041422 000137 042614 JMP 1$ ;NO - BYPASS REST OF TEST
6511 041426 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
6512 041434 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6513 041442 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6514 041446 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
6515 041454 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6516 041462 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
6517 041466 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6518 041474 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6519 041502 042737 077777 001164 BIC #ICATA, $TMP0 ;SAVE SPECIFIED BITS
6520 041510 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6521 041516 001414 BEQ 69$ ;BR IF OK
6522 041520 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6523 041526 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
6524 041534 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6525 041542 104010 ERROR 10 ;REPORT THE ERROR
6526 041544 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6527 041550 000240 69$: NOP
6528 041552 005737 001250 TST CKERR ;WAS ATTENTION SET FOR A??
6529 041556 001402 BEQ .+6 ;YES!!
6530 041560 000137 042614 JMP 1$ ;NO - BYPASS REST OF TEST
6531
6532 ;*****
6533
6534 ;SEIZE THE DRIVE THROUGH PORT B
6535
6536 041564 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
6537 041572 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
6538 041600 005060 000012 CLR RMDS1(RO) ;WRITE RMDS1
6539 041604 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
6540 041612 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6541 041620 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
6542 041626 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
6543 041634 010037 001122 MOV RO, $B0ADR ;RH11 BASE ADDRESS
6544 041640 062737 000012 001122 ADD #RMDS1, $B0ADR ;GENERATE BAD REGISTER ADDRESS
6545 041646 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
6546 041652 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
6547 041660 001403 BEQ 71$ ;BR IF IT IS
6548 041662 104004 ERROR 4 ;REPORT THE ERROR
6549 041664 000137 042614 JMP 1$ ;BYPASS REST OF THE SUBTEST

```

F10

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T31

MACY11 30(1046) 01-AUG-77 11:02 PAGE 122
 RESET ATTENTION 'B' BY GO TEST

SEQ 0124

```

6550 041670
6551 041670 113760 001226 000010
6552 041676 013737 001226 001240
6553 041704 016037 000012 001126
6554 041712 042737 020001 001126
6555 041720 012737 011700 001124
6556 041726 013737 001124 001166
6557 041734 005137 001166
6558 041740 013737 001126 001164
6559 041746 043737 001166 001164
6560 041754 023737 001124 001164
6561 041762 001401
6562 041764 104005
6563 041766 000240
6564
6565
6566
6567
6568
6569 041770 012760 000001 000000
6570
6571
6572
6573
6574 041776 005037 001250
6575 042002 016037 000012 001126
6576 042010 012737 000012 001122
6577 042016 060037 001122
6578 042022 005037 001124
6579 042026 013737 001126 001164
6580 042034 042737 077777 001164
6581 042042 023737 001124 001164
6582 042050 001414
6583 042052 013737 001126 001174
6584 042060 042737 100000 001174
6585 042066 053737 001174 001124
6586 042074 104061
6587 042076 005137 001250
6588 042102 000240
6589
6590
6591
6592
6593
6594 042104 113760 001226 000010
6595 042112 013737 001226 001240
6596 042120 012760 000013 000000
6597
6598
6599
6600 042126 005037 001254
6601 042132 012737 000012 001122
6602 042140 060037 001122
6603 042144 012737 011700 001124
6604 042152 113760 001224 000010
6605 042160 016037 000012 001170

```

```

71$:
MOV#B PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
BIC #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
BEQ 72$ ;BR IF THEY ARE
ERROR 5 ;REPORT THE ERROR
NOP

72$:
;*****
;ISSUE NOP COMMAND TO PORT B
MOV #1,RMCS1(RO)
;*****
;VERIFY THAT ATTENTION FOR PORT B CLEARED
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
MOV #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 73$ ;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
ERROR 61 ;TYPE MESSAGE 61
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

73$:
;*****
;RELEASE THE DRIVE FROM PORT B
MOV#B PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RMDS1,$BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
MOV#B PORTA, RMCS2(RO) ;SELECT PORT A.
MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.

```

G10

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T31

MACY11 30(1046) 01-AUG-77 11:02 PAGE 123
 RESET ATTENTION 'B' BY GO TEST

SEQ 0125

```

6606 042166 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
6607 042174 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6608 042202 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6609 042210 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
6610 042216 016037 000012 001172 MOV RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6611 042224 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
6612 04223 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
6613 042240 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6614 042246 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6615 042254 001006 BNE 75$ ;BR IF NOT
6616 042256 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6617 042262 001045 BNE 77$ ;BR IF NOT
6618 042264 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6619 042266 000137 042466 JMP 79$ ;BYPASS THE REST OF THE CHECKS
6620 042272 013737 001170 001126 75$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6621 042300 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6622 042306 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
6623 042314 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6624 042320 001414 BEQ 76$ ;BR IF ZERO
6625 042322 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6626 042330 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6627 042336 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
6628 042344 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6629 042350 001012 BNE 77$ ;BR IF NOT
6630 042352 012737 177777 001254 76$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6631 042360 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
6632 042366 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
6633 042374 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6634 042376 013737 001170 001126 77$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6635 042404 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6636 042412 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6637 042420 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
6638 042426 001401 BEQ 78$ ;BR IF OK FROM PORT A.
6639 042430 104007 ERROR 7 ;REPORT ERROR
6640 042432 013737 001172 001126 78$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6641 042440 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6642 042446 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6643 042454 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6644 042462 001401 BEQ 79$ ;BR IF OK
6645 042464 104007 ERROR 7 ;REPORT ERROR
6646 042466 000240 79$: NOP
6647
6648 ;*****
6649
6650 ;VERIFY THAT ATTENTION FOR PORT A IS STIL SET
6651
6652 042470 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
6653 042476 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6654 042504 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6655 042510 016037 000012 001126 MOV RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
6656 042516 012737 000012 001122 MOV #RMDS1,$BDDAR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6657 042524 060037 001122 ADD RO,$BDDAR ;ADD RH11 BASE ADDRESS
6658 042530 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6659 042536 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6660 042544 042737 077777 001164 BIC #CATA,$TMP0 ;SAVE SPECIFIED BITS
6661 042552 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
    
```

H10

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T31

MACY11 30(1046) 01-AUG-77 11:02 PAGE 124
RESET ATTENTION 'B' BY GO TEST

SEQ 0126

6662	042560	001414			BEQ	80\$;BR IF OK
6663	042562	013737	001126	001174	MOV	\$BDDAT,\$TMP4		;COPY 'BAD DATA'
6664	042570	042737	100000	001174	BIC	#ATA,\$TMP4		;CLEAR THE MASKED BITS
6665	042576	053737	001174	001124	BIS	\$TMP4,\$GDOAT		; 'OR' WITH GOOD DATA FOR TYPEOUT
6666	042604	104062			ERROR	62		;TYPE MESSAGE 62
6667	042606	005137	001250		COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
6668	042612	000240			80\$: NOP			
6669	042614	000004			1\$: SCOPE			

```

*****
*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. SFT 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.
*****

```

6684					TST	KYBCTL		;PERFORMING ONLY SINGLE TESTS ?
6685	042616				BEQ	2\$;BR IF NOT
6686	042616	005737	001300		1\$: BPL	1\$;BR IF JUST ENTERED TEST
6687	042622	001406			JMP	EXEC		;RETURN & GET NEXT TEST NUMBER
6688	042624	100002			1\$: MOV	#-1,KYBCTL		;SET SINGLE TEST INDICATOR
6689	042626	000137	002676		2\$: MOV	#32,\$STNM		;TEST NUMBER
6690	042632	012737	177777	001300	MOV	#TEST32,\$LPADR		;LOAD LOOP ON TEST ADDRESS
6691	042640	112737	000032	001102	MOV	#TEST32,\$LPERR		;LOAD LOOP ON ERROR ADDRESS
6692	042646	012737	042670	001106	MOV	#25,\$TIMES		;DO 25. ITERATIONS
6693	042654	012737	042670	001110	TEST32: MOV	#STACK,SP		;LOAD THE STACK POINTER
6694	042662	012737	000031	001176				
6695	042670	012706	001100					

```

*****
;SET ATTENTION BITS FOR BOTH PORTS
66$: MOVB  PORTA,RMCS2(RO) ;SELECT PORT 64$
TST  RMDS1(RO) ;MAKE SURE DRIVE AVAILABLE
BEQ  66$
MOV  #-1,RMER1(RO) ;FORCE ERRORS
CLR  RMER1(RO) ;CLEAR THE ERRORS
64$: MOV  PORTB,RMCS2(RO) ;SELECT THE OTHER PORT
TST  RMDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ  64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV  #-1,RMER1(RO) ;FORCE ERRORS ON PORT 65$
CLR  RMER1(RO) ;CLEAR THE ERRORS
65$: MOVB  PORTA,RMCS2(RO) ;SELECT PORT "64$" AGAIN
TST  RMDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ  65$ ;BR IF DRIVE HASN'T TIMED OUT

```

```

*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

6717	042764	113760	001224	000010	MOVB	PORTA,RMCS2(RO)		;SELECT PORT A
------	--------	--------	--------	--------	------	-----------------	--	----------------

```

6718 042772 013737 001224 001240      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6719 043000 005037 001250      CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
6720 043004 016037 000012 001126      MOV    RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
6721 043012 012737 000012 001122      MOV    #RMD51,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6722 043020 060037 001122      ADD    RO,SBADR    ;ADD RH11 BASE ADDRESS
6723 043024 012737 100000 001124      MOV    #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
6724 043032 013737 001126 001164      MOV    SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
6725 043040 042737 077777 001164      BIC    #1CATA,STMP0 ;SAVE SPECIFIED BITS
6726 043046 023737 001124 001164      CMP    SGDDAT,STMP0 ;COMPARE THE BITS
6727 043054 001414      BEQ    675        ;BR IF OK
6728 043056 013737 001126 001174      MOV    SBDDAT,STMP4 ;COPY 'BAD DATA'
6729 043064 042737 100000 001174      BIC    #ATA,STMP4  ;CLEAR THE MASKED BITS
6730 043072 053737 001174 001124      BIS    STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6731 043100 104010      ERROR  10        ;REPORT THE ERROR
6732 043102 005137 001250      COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
6733 043106 000240      NOP
6734 043110 005737 001250      TST    CKERR      ;WAS ATTN BIT FOR PORT A SET ?
6735 043114 001402      BEQ    .+6        ;BR IF IT WAS
6736 043116 000137 044072      JMP    IS        ;BYPASS REST OF TEST IF NOT
6737 043122 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
6738 043130 013737 001226 001240      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6739 043136 005037 001250      CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
6740 043142 016037 000012 001126      MOV    RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
6741 043150 012737 000012 001122      MOV    #RMD51,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6742 043156 060037 001122      ADD    RO,SBADR    ;ADD RH11 BASE ADDRESS
6743 043162 012737 100000 001124      MOV    #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
6744 043170 013737 001126 001164      MOV    SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
6745 043176 042737 077777 001164      BIC    #1CATA,STMP0 ;SAVE SPECIFIED BITS
6746 043204 023737 001124 001164      CMP    SGDDAT,STMP0 ;COMPARE THE BITS
6747 043212 001414      BEQ    695        ;BR IF OK
6748 043214 013737 001126 001174      MOV    SBDDAT,STMP4 ;COPY 'BAD DATA'
6749 043222 042737 100000 001174      BIC    #ATA,STMP4  ;CLEAR THE MASKED BITS
6750 043230 053737 001174 001124      BIS    STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6751 043236 104010      ERROR  10        ;REPORT THE ERROR
6752 043240 005137 001250      COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
6753 043244 000240      NOP
6754 043246 005737 001250      TST    CKERR      ;WAS ATTN BIT FOR PORT B SET ?
6755 043252 001402      BEQ    .+6        ;BR IF IT WAS
6756 043254 000137 044072      JMP    IS        ;BYPASS REST OF TEST IF NOT
6757
6758 ;*****
6759
6760 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6761
6762 043260 005037 001254      CLR    RELERR     ;CLEAR THE 'RELEASE ERROR' INDICATOR
6763 043264 012737 000012 001122      MOV    #RMD51,SBADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
6764 043272 060037 001122      ADD    RO,SBADR   ;ADD THE I/O BASE ADDRESS
6765 043276 012737 111700 001124      MOV    #111700,SGDDAT ;COMPARISON CONSTANT
6766 043304 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
6767 043312 016037 000012 001170      MOV    RMD51(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6768 043320 042737 024001 001170      BIC    #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
6769 043326 013737 001170 001164      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
6770 043334 042737 100100 001164      BIC    #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6771 043342 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
6772 043350 016037 000012 001172      MOV    RMD51(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6773 043356 042737 024001 001172      BIC    #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES

```

J10

MD-11-DZRMG-A, RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T32

MACY11 30(1046) 01-AUG-77 11:02 PAGE 126
TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT

SEQ 0128

```

6774 043364 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO 'TMP1'
6775 043372 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6776 043400 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6777 043406 001006                BNE      71$              ;BR IF NOT
6778 043410 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6779 043414 001045                BNE      73$              ;BR IF NOT
6780 043416 104046                ERROR    46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6781 043420 000137 043604                JMP      75$              ;BYPASS THE REST OF THE CHECKS
6782 043424 013737 001170 001126 71$:      MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6783 043432 013737 001226 001240                MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6784 043440 113760 001226 000010                MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B.
6785 043446 005737 001164                TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6786 043452 001414                BEQ      72$              ;BR IF ZERO
6787 043454 013737 001224 001240                MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6788 043462 013737 001172 001126                MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
6789 043470 113760 001224 000010                MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
6790 043476 005737 001166                TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6791 043502 001012                BNE      73$              ;BR IF NOT
6792 043504 012737 177777 001254 72$:      MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6793 043512 012760 000011 000000                MOV      #11,RMCS1(RO)   ;CLEAR THE DRIVE
6794 043520 012760 000013 000000                MOV      #13,RMCS1(RO)   ;RELEASE THE DRIVE
6795 043526 104026                ERROR    26              ;TYPE ERROR MESSAGE 26
6796 043530 013737 001170 001126 73$:      MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6797 043536 013737 001224 001240                MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
6798 043544 023737 001124 001126                CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
6799 043552 001401                BEQ      74$              ;BR IF OK FROM PORT A.
6800 043554 104007                ERROR    7               ;REPORT ERROR
6801 043556 013737 001172 001126 74$:      MOV      $TMP3,$BDDAT     ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6802 043564 013737 001226 001240                MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
6803 043572 023737 001124 001126                CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
6804 043600 001401                BEQ      75$              ;BR IF OK
6805 043602 104007                ERROR    7               ;REPORT ERNOR
6806 043604 000240                NOP                          ;
6807 043606 005737 001254                TST      RELERR          ;WAS DRIVE IN NEUTRAL ?
6808 043612 001402                BEQ      .+6              ;BR IF IT WAS
6809 043614 000137 044072                JMP      1$               ;BYPASS RESET OF TEST
6810                ;*****
6811                ;ISSUE THE MASSBUS INIT
6812
6813 043620 012760 000040 000010                MOV      #CLR,RMCS2(RO)  ;ISSUE A MASSBUS INIT
6814
6815                ;*****
6816                ;CHECK THE ATTENTION BITS OF BOTH PORTS
6817
6818 043626 113760 001224 000010                MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A
6819 043634 013737 001224 001240                MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6820 043642 005037 001250                CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6821 043646 016037 000012 001126                MOV      RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
6822 043654 012737 000012 001122                MOV      #RMDS1,$BDDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6823 043662 060037 001122                ADD      RO,$BDDADR      ;ADD RH11 BASE ADDRESS
6824 043666 005037 001124                CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
6825 043672 013737 001126 001164                MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO 'TMP0'
6826 043700 042737 077777 001164                BIC      #!CATA,$TMP0    ;SAVE SPECIFIED BITS
6827 043706 023737 001124 001164                CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
6828 043714 001414                BEQ      76$              ;BR IF OK
6829 043716 013737 001126 001174                MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'

```


K10

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T32

MACY11 30(1046) 01-AUG-77 11:02 PAGE 127
TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT

SEQ 0129

6830	043724	042737	100000	001174	BIC	#ATA,STMP4	;CLEAR THE MASKED BITS
6831	043732	053737	001174	001124	BIS	STMP4,\$GDDAT	;'OR' WITH GOOD DATA FOR TYPEOUT
6832	043740	104051			ERROR	51	;TYPE MESSAGE 51
6833	043742	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6834	043746	000240			76\$: NOP		
6835	043750	113760	001126	000010	MOV	PORTB, RMCS2(RO)	;SELECT PORT B
6836	043756	013737	001226	001240	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6837	043764	005037	001250		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
6838	043770	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
6839	043776	012737	000012	001122	MOV	#RMDS1, \$BADDR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
6840	044004	060037	001122		ADD	RO, \$BADDR	;ADD RH11 BASE ADDRESS
6841	044010	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
6842	044014	013737	001126	001164	MOV	\$BDDAT, \$TMPD	;MOVE REGISTER CONTENTS TO 'STMPD'
6843	044022	042737	077777	001164	BIC	#ICATA, \$TMPD	;SAVE SPECIFIED BITS
6844	044030	023737	001124	001164	CMP	\$GDDAT, \$TMPD	;COMPARE THE BITS
6845	044036	001414			BEQ	78\$;BR IF OK
6846	044040	013737	001126	001174	MOV	\$BDDAT, \$TMP4	;COPY 'BAD DATA'
6847	044046	042737	100000	001174	BIC	#ATA, \$TMP4	;CLEAR THE MASKED BITS
6848	044054	053737	001174	001124	BIS	STMP4, \$GDDAT	;'OR' WITH GOOD DATA FOR TYPEOUT
6849	044062	104051			ERROR	51	;TYPE MESSAGE 51
6850	044064	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6851	044070	000240			78\$: NOP		
6852	044072	000004			1\$: SCOPE		;LOOP ?

```

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

```

6870	044074				†ST33:		
6871	044074	005737	001300		IST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
6872	044100	001406			BEQ	2\$;BR IF NOT
6873	044102	100002			BPL	1\$;BR IF JUST ENTERED TEST
6874	044104	000137	002676		JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
6875	044110	012737	177777	001300	1\$: MOV	#-1, KYBCTL	;SET SINGLE TEST INDICATOR
6876	044116	112737	000033	001102	2\$: MOV	#33, \$STNM	;TEST NUMBER
6877	044124	012737	044146	001106	MOV	#TEST33, \$LPADR	;LOAD LOOP ON TEST ADDRESS
6878	044132	012737	044146	001110	MOV	#TEST33, \$LPERR	;LOAD LOOP ON ERROR ADDRESS
6879	044140	012737	000002	001176	MOV	#2, \$TIMES	;DO 2. ITERATIONS
6880	044146	012706	001100		TEST33: MOV	#STACK, SP	;LOAD THE STACK POINTER
6881					;;		*****
6882							
6883							;SET ATTENTION BITS FOR BOTH PORTS
6884							
6885	044152	113760	001224	000010	MOV	PORTA, RMCS2(RO)	;SELECT PORT 64\$

L10

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T33

MACY11 30(1046) 01-AUG-77 11:02 PAGE 128
RESET ATTENTION 'A' & 'B' BY RMAS

SEQ 0130

```

6886 044160 005760 000012 66$: TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
6887 044164 001775 BEQ 66$
6888 044166 012760 177777 000014 MOV #-1,RMER1(RO) ;FORCE ERRORS
6889 044174 005060 000014 CLR RMER1(RO) ;CLEAR THE ERRORS
6890 044200 013760 001226 000010 MOV PORTB,RMCS2(RO) ;SELECT THE OTHER PORT
6891 044206 005760 000012 64$: TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6892 044212 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
6893 044214 012760 177777 000014 MOV #-1,RMER1(RO) ;FORCE ERRORS ON PORT 65$
6894 044222 005060 000014 CLR RMER1(RO) ;CLEAR THE ERRORS
6895 044226 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT "64$" AGAIN
6896 044234 005760 000012 65$: TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6897 044240 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
6898 ;*****
6899 ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
6900 044242 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
6901 044250 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6902 044256 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6903 044262 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
6904 044270 012737 000012 001122 MOV #RMD51,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6905 044276 060037 001122 ADD RO,SBDADR ;ADD RHI1 BASE ADDRESS
6906 044302 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6907 044310 013737 001126 001164 MOV SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6908 044316 042737 077777 001164 BIC #+CATA,$TMP0 ;SAVE SPECIFIED BITS
6909 044324 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6910 044332 001414 BEQ 67$ ;BR IF OK
6911 044334 013737 001126 001174 MOV SBDAT,$TMP4 ;COPY 'BAD DATA'
6912 044342 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6913 044350 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6914 044356 104010 ERROR 10 ;REPORT THE ERROR
6915 044360 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6916 044364 000240 67$: NOP
6917 044366 005737 001250 TST CKERR ;WAS ATA SET FOR A??
6918 044372 001402 BEQ .+6 ;YES - CONTINUE
6919 044374 000137 045216 JMP 1$ ;BYPASS REST OF TEST
6920 044400 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
6921 044406 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6922 044414 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6923 044420 016037 000012 001126 MOV RMD51(RO),SBDAT ;GET CONTENTS OF RMD51
6924 044426 012737 000012 001122 MOV #RMD51,SBDADR ;FORM REGISTER ADDRESS OF R MESSAGE
6925 044434 060037 001122 ADD RO,SBDADR ;ADD RHI1 BASE ADDRESS
6926 044440 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6927 044446 013737 001126 001164 MOV SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6928 044454 042737 077777 001164 BIC #+CATA,$TMP0 ;SAVE SPECIFIED BITS
6929 044462 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6930 044470 001414 BEQ 69$ ;BR IF OK
6931 044472 013737 001126 001174 MOV SBDAT,$TMP4 ;COPY 'BAD DATA'
6932 044500 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6933 044506 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6934 044514 104010 ERROR 10 ;REPORT THE ERROR
6935 044516 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6936 044522 000240 69$: NOP
6937 044524 005737 001250 TST CKERR ;WAS ATA SET FOR B??
6938 044530 001402 BEQ .+6 ;YES - CONTINUE
6939 044532 000137 045216 JMP 1$ ;BYPASS REST OF TEST
6940 ;*****
6941 ;*****

```

M10

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P1! 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 129
T33 RESET ATTENTION 'A' & 'B' BY RMAS

SEQ 0131

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

6942
6943
6944 044536 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
6945 044542 012737 000012 001122 MOV #RMS1,$BODR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
6946 044550 060037 001122 ADD R0,$BODR ;ADD THE I/O BASE ADDRESS
6947 044554 012737 111700 001124 MOV #11700,$GDDAT ;COMPARISON CONSTANT
6948 044562 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
6949 044570 016037 000012 001170 MOV RMS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6950 044576 042737 024001 001170 BIC #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
6951 044604 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6952 044612 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6953 044620 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
6954 044626 016037 000012 001172 MOV RMS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6955 044634 042737 024001 001172 BIC #PIP!WAL!OM,$TMP3 ;CLEAR DONT CARES
6956 044642 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
6957 044650 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6958 044656 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6959 044664 001006 BNE 71$ ;BR IF NOT
6960 044666 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6961 044672 001045 BNE 73$ ;BR IF NOT
6962 044674 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6963 044676 000137 045062 JMP 75$ ;BYPASS THE REST OF THE CHECKS
6964 044702 013737 001170 001126 71$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6965 044710 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6966 044716 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
6967 044724 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6968 044730 001414 BEQ 72$ ;BR IF ZERO
6969 044732 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6970 044740 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6971 044746 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
6972 044754 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6973 044760 001012 BNE 73$ ;BR IF NOT
6974 044762 012737 177777 001254 72$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6975 044770 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
6976 044776 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
6977 045004 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6978 045006 013737 001170 001126 73$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
6979 045014 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6980 045022 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
6981 045030 001401 BEQ 74$ ;BR IF OK FROM PORT A.
6982 045032 104007 ERROR 7 ;REPORT ERROR
6983 045034 013737 001172 001126 74$: MOV $TMP3,$BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
6984 045042 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6985 045050 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6986 045056 001401 BEQ 75$ ;BR IF OK
6987 045060 104007 ERROR 7 ;REPORT ERROR
6988 045062 000240 75$: NOP
6989 045064 005737 001254 TST RELERR ;WAS DRIVE IN NEUTRAL??
6990 045070 001402 BEQ .+6 ;YES!!
6991 045072 000137 045216 JMP 1$ ;BYPASS REST OF TEST
6992
6993 ;*****
6994 ;WRITE THE ATTENTION BIT
6995 045076 013760 001236 000016 MOV ASR1,RMAS(R0)
6996
6997 ;VERIFY THAT BOTH ATTENTIONS ARE RESET BY READING RMAS

```

RESET ATTENTION 'A' & 'B' BY RMAS

```

6998 045104 016037 000016 001126 MOV RMAS(RO),SBDAT ;GET ATTENTION SUMMARY
6999 045112 033737 001236 001126 BIT ASR1,SBDAT ;IS THE ATTENTION RESET ??
7000 045120 001414 BEQ 2$ ;YES !!
7001 045122 010037 001122 MOV RO,SBDADR ;SETUP REGISTER ADDRESS
7002 045126 062737 000016 001122 ADD #RMAS,SBDADR
7003 045134 013737 001126 001124 MOV SBDAT,$GDAT ;SETUP EXPECTED DATA
7004 045142 043737 001236 001124 BIC ASR1,$GDAT ;RESET THIS DRIVES BIT
7005 045150 104060 ERROR 60 ;ATTENTION NOT CLEARED BY RMAS

```

```

7006
7007 045152 2$:
7008 ;WAIT FOR THE DRIVES TO RELEASE BY TIMEOUT
7009

```

```

7010 045152 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
7011 045160 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7012 045166 005760 000012 3$: TST RMD51(RO) ;MAKE SURE DEVICE IS AVAILABLE
7013 045172 001775 BEQ 3$
7014 045174 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
7015 045202 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7016 045210 005760 000012 4$: TST RMD51(RO) ;MAKE SURE DEVICE IS AVAILABLE
7017 045214 001775 BEQ 4$
7018 045216 000004 1$: SCOPE
7019
7020

```

```

*****
*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 RMD51.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.
*****

```

```

7034 045220 005737 00130. 1$: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
7035 045220 001406 BEQ 2$ ;BR IF NOT
7036 045224 100002 BPL 1$ ;BR IF JUST ENTERED TEST
7037 045226 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
7038 045230 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
7039 045234 112737 000034 001102 2$: MOVB #34,$STN ;TEST NUMBER
7040 045242 012737 045272 001106 MOV #TEST34,$LPADR ;LOAD LOOP ON TEST ADDRESS
7041 045250 012737 045272 001110 MOV #TEST34,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7042 045256 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
7043 045264 012706 001100 TEST34: MOV #STACK,SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

7044
7045
7046
7047
7048 045276 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
7049 045304 005060 000012 CLR RMD51(RO) ;SEIZE THE DRIVE
7050 045310 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7051 045316 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
7052 045324 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
7053 045332 005060 000012 CLR RMD51(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'

```

7054	045336	012760	000011	000000	MOV	#11, RMCS1(RO)	;ISSUE DRIVE CLEAR
7055	045344	012760	000013	000000	MOV	#13, RMCS1(RO)	;RELEASE THE DRIVE
7056	045352	113760	001224	000010	MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
7057	045360	012760	177777	000014	MOV	#-1, RMER1(RO)	;SET ERRORS TO FORCE ATTN BIT ON PORT A
7058	045366	005060	000014		CLR	RMER1(RO)	;CLEAR THE ERRORS
7059	045372	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
7060	045400	005760	000012		1\$: TST	RMDS1(RO)	;WAIT FOR DRIVE TO RETURN TO NEUTRAL
7061	045404	001775			BEQ	1\$;BR IF STILL SEIZED BY PORT A
7062	045406	012737	000016	001122	MOV	#RMAS, \$B0ADR	;FORM ADDRESS OF ATTN REG IF ERROR
7063	045414	060037	001122		ADD	RO, \$B0ADR	;ADD THE ADDRESS BASE
7064	045420	013737	001236	001124	MOV	ASR1, \$GDDAT	;GOOD DATA FOR ERROR MESSAGE
7065	045426	013737	001236	001166	MOV	ASR1, \$TMP1	;MAKE DATA COMPARE MASK
7066	045434	005137	001166		COM	\$TMP1	;COMPLEMENT IT
7067	045440	012737	045474	001110	MOV	#2\$, \$LPERR	;LOAD LOOP ON ERROR ADDRESS
7068	045446	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
7069	045454	013737	001226	001240	MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7070	045462	013737	001226	001242	MOV	PORTB, SEIZPT	; 'SEIZED' PORT ADDRESS
7071	045470	005060	000012		CLR	RMDS1(RO)	;SEIZE THE DRIVE THROUGH PORT B
7072	045474	016037	000016	001126	2\$: MOV	RMAS(RO), \$B0DAT	;GET THE CONTENTS OF THE ATTENTION REG
7073	045502	013737	001126	001164	MOV	\$B0DAT, \$TMP0	;PUT CONTENTS INTO WORKING LOCATION
7074	045510	043737	001166	001164	BIC	\$TMP1, \$TMP0	;CLEAR OTHER BITS
7075	045516	023737	001124	001164	CMP	\$GDDAT, \$TMP0	;SEE IF ATTN BIT FOR DRIVE SET
7076	045524	001401			BEQ	3\$;BR IF SET
7077	045526	104053			ERROR	53	;REPORT THE ERROR
7078	045530				3\$:		
7079							
7080							
7081							;RELEASE THE DRIVE FROM PORT B
7082	045530	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
7083	045536	013737	001226	001240	MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7084	045544	012760	000013	000000	MOV	#13, RMCS1(RO)	;ISSUE RELEASE THROUGH PORT B
7085							
7086							;VERIFY THAT THE DRIVE IS IN NEUTRAL
7087							
7088	045552	005037	001254		CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
7089	045556	012737	000012	001122	MOV	#RMDS1, \$B0ADR	;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7090	045564	060037	001122		ADD	RO, \$B0ADR	;ADD THE I/O BASE ADDRESS
7091	045570	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;COMPARISON CONSTANT
7092	045576	113760	001224	000010	MOVB	PORTA, RMCS2(RO)	;SELECT PORT A.
7093	045604	016037	000012	001170	MOV	RMDS1(RO), \$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
7094	045612	042737	024001	001170	BIC	#PIP!WRL!OM, \$TMP2	;CLEAR DONT CARES
7095	045620	013737	001170	001164	MOV	\$TMP2, \$TMP0	;COPY IT INTO 'TMP0'
7096	045626	042737	100100	001164	BIC	#ATA!VV, \$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7097	045634	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B.
7098	045642	016037	000012	001172	MOV	RMDS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
7099	045650	042737	024001	001172	BIC	#PIP!WRL!OM, \$TMP3	;CLEAR DONT CARES
7100	045656	013737	001172	001166	MOV	\$TMP3, \$TMP1	;COPY IT INTO 'TMP1'
7101	045664	042737	100100	001166	BIC	#ATA!VV, \$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7102	045672	023737	001164	001166	CMP	\$TMP0, \$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7103	045700	001006			BNE	64\$;BR IF NOT
7104	045702	005737	001164		TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7105	045706	001045			BNE	66\$;BR IF NOT
7106	045710	104046			ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7107	045712	000137	046112		JMP	68\$;BYPASS THE REST OF THE CHECKS
7108	045716	013737	001170	001126	64\$: MOV	\$TMP2, \$B0DAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7109	045724	013737	001226	001240	MOV	PORTB, PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

7110 045732 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ;SELECT PORT B.
7111 045740 005737 001164              TST   $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7112 045744 001414              BEQ   65$             ;BR IF ZERO
7113 045746 013737 001224 001240      MOV   PORTA, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7114 045754 013737 001172 001126      MOV   $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
7115 045762 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT A.
7116 045770 005737 001166              TST   $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7117 045774 001012              BNE   66$             ;BR IF NOT
7118 045776 012737 177777 001254 65$:  MOV   #-1, RELERR     ;SET 'RELEASE ERROR' INDICATOR
7119 046004 012760 000011 000000      MOV   #11, RMCS1(RO)  ;CLEAR THE DRIVE
7120 046012 012760 000013 000000      MOV   #13, RMCS1(RO)  ;RELEASE THE DRIVE
7121 046020 104026              ERROR 26              ;TYPE ERROR MESSAGE 26
7122 046022 013737 001170 001126 66$:  MOV   $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7123 046030 013737 001224 001240      MOV   PORTA, PTNBR    ;CHANGE PORT NUMBER
7124 046036 042737 100000 001126      BIC   #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
7125 046044 023737 001124 001126      CMP   $GDDAT, $BDDAT  ;ALL BITS OK ?
7126 046052 001401              BEQ   67$             ;BR IF OK FROM PORT A.
7127 046054 104007              ERROR 7               ;REPORT ERROR
7128 046056 013737 001172 001126 67$:  MOV   $TMP3, $BDDAT   ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7129 046064 013737 001226 001240      MOV   PORTB, PTNBR    ;CHANGE PORT NUMBER
7130 046072 042737 100000 001126      BIC   #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
7131 046100 023737 001124 001126      CMP   $GDDAT, $BDDAT  ;SEE IF READ OK FROM PORT B.
7132 046106 001401              BEQ   68$             ;BR IF OK
7133 046110 104007              ERROR 7               ;REPORT ERROR
7134 046112 000240 68$:  NOP
7135 046114 000004              SCOPE                 ;LOOP ?

```

```

7136
7137 ;*****
7138 ;*TEST 35      PCRT 'B' ALTERNATE ATTENTION PATH TEST
7139 ;*
7140 ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
7141 ;*
7142 ;*  A.  SET THE ATTENTION BIT FOR PORT 'B'.
7143 ;*
7144 ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
7145 ;*
7146 ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
7147 ;*      FOR THE DRIVE IS SET.
7148 ;*
7149 ;*****

```

```

7150 046116 005737 001300      †ST35:
7151 046116 001406              TST   KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7152 046122 100002              BEQ   2$              ;BR IF NOT
7153 046124 000137 002676      BPL   1$              ;BR IF JUST ENTERED TEST
7154 046132 012737 177777 001300 1$:  JMP   EXEC            ;RETURN & GET NEXT TEST NUMBER
7155 046140 112737 000035 001102 2$:  MOV   #-1, KYBCTL     ;SET SINGLE TEST INDICATOR
7156 046146 012737 046170 001106      MOVB  #35, $TSTNM     ;TEST NUMBER
7157 046154 012737 046170 001110      MOV   #TEST35, $LPADR ;LOAD LOOP ON TEST ADDRESS
7158 046162 012737 000031 001176      MOV   #TEST35, $LPERR ;LOAD LOOP ON ERROR ADDRESS
7159 046170 012706 001100      MOV   #25, $TIMES     ;DO 25. ITERATIONS
7160              TEST35: MOV   #STACK, SP      ;LOAD THE STACK POINTER
7161              ;CLEAR ATTENTION BITS FOR BOTH PORTS
7162
7163
7164 046174 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT #A
7165 046202 005060 000012      CLR   RMDS1(RO)       ;SEIZE THE DRIVE

```

```

7166 046206 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7167 046214 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
7168 046222 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT #B
7169 046230 005060 000012 CLR RMD51(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
7170 046234 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7171 046242 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
7172 046250 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B
7173 046256 012760 177777 000014 MOV #-1,RMER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT B
7174 046264 005060 000014 CLR RMER1(RO) ;CLEAR THE ERRORS
7175 046270 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A
7176 046276 005760 000012 15: TST RMD51(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
7177 046302 001775 BEQ 15 ;BR IF STILL SEIZED BY PORT B
7178 046304 012737 000016 001122 MOV #RMAS,$B0ADR ;FORM ADDRESS OF ATTN REG IF ERROR
7179 046312 060037 001122 ADD RO,$B0ADR ;ADD THE ADDRESS BASE
7180 046316 013737 001236 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
7181 046324 013737 001236 001166 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
7182 046332 005137 001166 COM $TMP1 ;COMPLEMENT IT
7183 046336 012737 046372 001110 MOV #25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7184 046344 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A
7185 046352 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7186 046360 013737 001224 001242 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
7187 046366 005060 000012 CLR RMD51(RO) ;SEIZE THE DRIVE THROUGH PORT A
7188 046372 016037 000016 001126 25: MOV RMAS(RO),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
7189 046400 013737 001126 001164 MOV $BDDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
7190 046406 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
7191 046414 023737 001124 001164 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
7192 046422 001401 BEQ 35 ;BR IF SET
7193 046424 104053 ERROR 53 ;REPORT THE ERROR
7194 046426 35:
7195
7196 ;RELEASE THE DRIVE FROM PORT A
7197
7198 046426 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A
7199 046434 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7200 046442 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7201
7202 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7203
7204 046450 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7205 046454 012737 000012 001122 MOV #RMD51,$B0ADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
7206 046462 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
7207 046466 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7208 046474 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A.
7209 046502 016037 000012 001170 MOV RMD51(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7210 046510 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7211 046516 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
7212 046524 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7213 046532 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B.
7214 046540 016037 000012 001172 MOV RMD51(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7215 046546 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7216 046554 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
7217 046562 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7218 046570 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7219 046576 001006 BNE 645 ;BR IF NOT
7220 046600 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7221 046604 001045 BNE 665 ;BR IF NOT

```

```

7222 046606 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7223 046610 000137 047010    JMP      68$      ;BYPASS THE REST OF THE CHECKS
7224 046614 013737 001170    64$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7225 046622 013737 001226 001240    MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7226 046630 113760 001226 000010    MOV     PORTB,RMCS2(RO) ;SELECT PORT B.
7227 046636 005737 001164          TST     $TMP0     ;SEE IF STATUS EQ 0 FROM PORT A.
7228 046642 001414          BEQ     65$      ;BR IF ZERO
7229 046644 013737 001224 001240    MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7230 046652 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7231 046660 113760 001224 000010    MOV     PORTA,RMCS2(RO) ;SELECT PORT A.
7232 046666 005737 001166          TST     $TMP1     ;SEE IF STATUS EQ ZERO FROM PORT B.
7233 046672 001012          BNE     66$      ;BR IF NOT
7234 046674 012737 177777 001254    65$:  MOV     #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
7235 046702 012760 000011 000000    MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
7236 046710 012760 000013 000000    MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
7237 046716 104026          ERROR 26          ;TYPE ERROR MESSAGE 26
7238 046720 013737 001170 001126    66$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMO51 READ
7239 046726 013737 001224 001240    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
7240 046734 042737 100000 001126    BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
7241 046742 023737 001124 001126    CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
7242 046750 001401          BEQ     67$      ;BR IF OK FROM PORT A.
7243 046752 104007          ERROR 7           ;REPORT ERROR
7244 046754 013737 001172 001126    67$:  MOV     $TMP3,$BDDAT ;CHECK RMO51 FOR BIT FAILURES - FROM PORT B.
7245 046762 013737 001226 001240    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
7246 046770 042737 100000 001126    BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
7247 046776 023737 001124 001126    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
7248 047004 001401          BEQ     68$      ;BR IF OK
7249 047006 104007          ERROR 7           ;REPORT ERROR
7250 047010 000240          68$:  NOP
7251 047012 000004          SCOPE ;LOOP ?

```

;;*****

```

*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.

```

```

7273 047014          †ST36:  TST     KYBCTL     ;PERFORMING ONLY SINGLE TESTS ?
7274 047014 005737 001300    BEQ     2$        ;BR IF NOT
7275 047020 001406          BPL     1$        ;BR IF JUST ENTERED TEST
7276 047022 100002          JMP     EXEC      ;RETURN & GET NEXT TEST NUMBER
7277 047024 000137 002676

```


F11

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T36

MACY11 30(1046) 01-AUG-77 11:02 PAGE 135
 SET ATTENTION 'A' BY COMMAND TEST

SEQ 0137

```

7278 047030 012737 177777 001300 15:  MOV    #1,KYBCTL      ;SET SINGLE TEST INDICATOR
7279 047036 112737 000036 001102 25:  MOVVB  #36,$TSTNM     ;TEST NUMBER
7280 047044 012737 047066 001106      MOV    #TEST36,$LPADR ;LOAD LOOP ON TEST ADDRESS
7281 047052 012737 047066 001110      MOV    #TEST36,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7282 047060 012737 000031 001176      MOV    #25,$TIMES     ;DO 25. ITERATIONS
7283 047066 012706 001100      TEST36: MOV   #STACK,$P  ;LOAD THE STACK POINTER
7284
7285      ;CLEAR ATTENTION BITS FOR BOTH PORTS
7286
7287 047072 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT #A
7288 047100 005060 000012      CLR    RMDS1(RO)      ;SEIZE THE DRIVE
7289 047104 012760 000011 000000      MOV    #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
7290 047112 012760 000013 000000      MOV    #13,RMCS1(RO)  ;RELEASE THE DRIVE
7291 047120 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT #B
7292 047126 005060 000012      CLR    RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
7293 047132 012760 000011 000000      MOV    #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
7294 047140 012760 000013 000000      MOV    #13,RMCS1(RO)  ;RELEASE THE DRIVE
7295 047146 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
7296 047154 013737 001224 001240      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7297 047162 013737 001224 001242      MOV    PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
7298
7299      ;*****
7300      ;DO A OFFSET THROUGH PORT A
7301
7302 047170 012760 000015 000000      MOV    #15,RMCS1(RO)  ;ISSUE A OFFSET INSTRUCTION THROUGH PORT A
7303
7304      ;*****
7305      ;WAIT FOR DRIVE TO FINISH
7306
7307 047176 032760 000200 000012      BIT    #DRY,RMDS1(RO) ;WAIT FOR DRIVE TO FINISH
7308 047204 001774      BEQ    #-6           ;BR IF NOT FINISHED
7309
7310      ;*****
7311      ;CONFIRM THAT ATTENTION IS SET FOR PORT A
7312
7313 047206 005037 001250      CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
7314 047212 016037 000012 001126      MOV    RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
7315 047220 012737 000012 001122      MOV    #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7316 047226 060037 001122      ADD    RO,$BDDADR     ;ADD RH11 BASE ADDRESS
7317 047232 012737 100000 001124      MOV    #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
7318 047240 013737 001126 001164      MOV    $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
7319 047246 042737 077777 001164      BIC    #1CATA,$STMP0 ;SAVE SPECIFIED BITS
7320 047254 023737 001124 001164      CMP    $GDDAT,$STMP0 ;COMPARE THE BITS
7321 047262 001414      BEQ    64$          ;BR IF OK
7322 047264 013737 001126 001174      MOV    $BDDAT,$STMP4 ;COPY 'BAD DATA'
7323 047272 042737 100000 001174      BIC    #ATA,$STMP4   ;CLEAR THE MASKED BITS
7324 047300 053737 001174 001124      BIS    $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7325 047306 104032      ERROR  32           ;TYPE MESSAGE 32
7326 047310 005137 001250      COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
7327 047314 000240      64$:  NOP
7328
7329      ;RELEASE THE DRIVE FROM PORT A
7330
7331 047316 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
7332 047324 013737 001224 001240      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7333 047332 012760 000013 000000      MOV    #13,RMCS1(RO)  ;ISSUE RELEASE THROUGH PORT A
  
```

G11

```

7334
7335
7336
7337 047340 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7338 047344 012737 000012 MOV #RMDS1,$BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7339 047352 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
7340 047356 012737 011700 MOV #MOL!PGM!OPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7341 047364 113760 001224 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
7342 047372 016037 000012 MOV RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7343 047400 042737 024001 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7344 047406 013737 001170 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
7345 047414 042737 100100 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7346 047422 113760 001226 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
7347 047430 016037 000012 MOV RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7348 047436 042737 024001 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7349 047444 013737 001172 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7350 047452 042737 100100 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7351 047460 023737 001164 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7352 047466 001006 BNE 66$ ;BR IF NOT
7353 047470 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7354 047474 001045 BNE 68$ ;BR IF NOT
7355 047476 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7356 047500 000137 047664 JMP 70$ ;BYPASS THE REST OF THE CHECKS
7357 047504 013737 001170 MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7358 047512 013737 001226 MOVB PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7359 047520 113760 001226 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
7360 047526 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7361 047532 001414 BEQ 67$ ;BR IF ZERO
7362 047534 013737 001224 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7363 047542 013737 001172 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7364 047550 113760 001224 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
7365 047556 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7366 047562 001012 BNE 68$ ;BR IF NOT
7367 047564 012737 177777 MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7368 047572 012760 000011 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
7369 047600 012760 000013 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
7370 047606 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
7371 047610 013737 001170 MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7372 047616 013737 001224 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7373 047624 023737 001124 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
7374 047632 001401 BEQ 69$ ;BR IF OK FROM PORT A.
7375 047634 104007 ERROR 7 ;REPORT ERROR
7376 047636 013737 001172 MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7377 047644 013737 001226 MOVB PORTB,PTNBR ;CHANGE PORT NUMBER
7378 047652 023737 001124 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
7379 047660 001401 BEQ 70$ ;BR IF OK
7380 047662 104007 ERROR 7 ;REPORT ERROR
7381 047664 000240 NOP ;
7382 047666 113760 001226 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
7383 047674 013737 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7384
7385 ;*****
7386 ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
7387
7388 047702 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7389 047706 016037 000012 MOV RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
  
```

H11

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 137
T36 SET ATTENTION 'A' BY COMMAND TEST

SEQ 0139

7390 047714 012737 000012 001122
7391 047722 060037 001122
7392 047726 005037 001124
7393 047732 013737 001126 001164
7394 047740 042737 077777 001164
7395 047746 023737 001124 001164
7396 047754 001414
7397 047756 013737 001126 001174
7398 047764 042737 100000 001174
7399 047772 053737 001174 001124
7400 050000 104032
7401 050002 005137 001250
7402 050006 000240
7403
7404
7405 050010 000004
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423 050012
7424 050012 005737 001300
7425 050016 001406
7426 050020 100002
7427 050022 000137 002676
7428 050026 012737 177777 001300
7429 050034 112737 000037 001102
7430 050042 012737 050064 001106
7431 050050 012737 050064 001110
7432 050056 012737 000031 001176
7433 050064 012706 001100
7434
7435
7436
7437 050070 113760 001224 000010
7438 050076 005060 000012
7439 050102 012760 000011 000000
7440 050110 012760 000013 000000
7441 050116 113760 001226 000010
7442 050124 005060 000012
7443 050130 012760 000011 000000
7444 050136 012760 000013 000000
7445 050144 113760 001226 000010

```

MOV #RMS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $B0ADR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $B0DAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 715 ;BR IF OK
MOV $B0DAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 32 ;TYPE MESSAGE 32
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
715:
;*****
; SCOPE ;LOOP ?
;*****
*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.
*
;*****
†ST37:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
25: MOV #37, $STNM ;TEST NUMBER
MOV #TEST37, $LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST37, $LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25, $TIMES ;DO 25. ITERATIONS
TEST37: MOV #STACK, SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOV#B PORTA, RMCS2(R0) ;SELECT PORT #A
CLR RMS1(R0) ;SEIZE THE DRIVE
MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
MOV#B PORTB, RMCS2(R0) ;SELECT PORT #B
CLR RMS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
MOV#B PORTB, RMCS2(R0) ;SELECT PORT B

```

```

7446 050152 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7447 050160 013737 001226 001242      MOV      PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
7448
7449                                     ;:*****
7450                                     ;DO A OFFSET THROUGH PORT B
7451
7452 050166 012760 000015 000000      MOV      #15,RMCS1(RO) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT B
7453
7454                                     ;:*****
7455                                     ;WAIT FOR DRIVE TO FINISH
7456
7457 050174 032760 000200 000012      BIT      #DRY,RMDS1(RO) ;WAIT FOR DRIVE TO FINISH
7458 050202 001774                      BEQ      .-6           ;BR IF NOT FINISHED
7459
7460                                     ;:*****
7461                                     ;CONFIRM THAT ATTENTION IS SET FOR PORT B
7462
7463 050204 005037 001250                      CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
7464 050210 016037 000012 001126      MOV      RMDS1(RO),SBDAT ;GET CONTENTS OF RMDS1
7465 050216 012737 000012 001122      MOV      #RMDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7466 050224 060037 001122                      ADD      RO,SBDADR    ;ADD RHI1 BASE ADDRESS
7467 050230 012737 100000 001124      MOV      #ATA,SDDAT ;WHAT REGISTER SHOULD BE
7468 050236 013737 001126 001164      MOV      SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7469 050244 042737 077777 001164      BIC      #CATA,$TMP0 ;SAVE SPECIFIED BITS
7470 050252 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
7471 050260 001414                      BEQ      64$         ;BR IF OK
7472 050262 013737 001126 001174      MOV      SBDAT,$TMP4 ;COPY 'BAD DATA'
7473 050270 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
7474 050276 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7475 050304 104032                      ERROR   32          ;TYPE MESSAGE 32
7476 050306 005137 001250                      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
7477 050312 000240                      64$: NOP
7478
7479                                     ;RELEASE THE DRIVE FROM PORT B
7480
7481 050314 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
7482 050322 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7483 050330 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
7484
7485                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7486
7487 050336 005037 001254                      CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
7488 050342 012737 000012 001122      MOV      #RMDS1,SBDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7489 050350 060077 001122                      ADD      RO,SBDADR    ;ADD THE I/O BASE ADDRESS
7490 050354 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7491 050362 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
7492 050370 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7493 050376 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7494 050404 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
7495 050412 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7496 050420 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
7497 050426 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7498 050434 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7499 050442 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7500 050450 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7501 050456 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```

```

7502 050464 001006          BNE      66$          ;BR IF NOT
7503 050466 005737 001164   TST      $TMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7504 050472 001045          BNE      68$          ;BR IF NOT
7505 050474 104046          ERROR    46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7506 050476 000137 050662   JMP      70$          ;BYPASS THE REST OF THE CHECKS
7507 050502 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7508 050510 013737 001226 001240   MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7509 050516 113760 001226 000010   MOV     PORTB,RMCS2(RO) ;SELECT PORT B.
7510 050524 005737 001164          TST      $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
7511 050530 001414          BEQ     67$          ;BR IF ZERO
7512 050532 013737 001224 001240   MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7513 050540 013737 001172 001126   MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7514 050546 113760 001224 000010   MOV     PORTA,RMCS2(RO) ;SELECT PORT A.
7515 050554 005737 001166          TST      $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
7516 050560 001012          BNE      68$          ;BR IF NOT
7517 050562 012737 177777 001254 67$:  MOV     #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
7518 050570 012760 000011 000000   MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
7519 050576 012760 000013 000000   MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
7520 050604 104026          ERROR    26          ;TYPE ERROR MESSAGE 26
7521 050606 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7522 050614 013737 001224 001240   MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
7523 050622 023737 001124 001126   CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
7524 050630 001401          BEQ     69$          ;BR IF OK FROM PORT A.
7525 050632 104007          ERROR    7           ;REPORT ERROR
7526 050634 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7527 050642 013737 001226 001240   MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
7528 050650 023737 001124 001126   CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
7529 050656 001401          BEQ     70$          ;BR IF OK
7530 050660 104007          ERROR    7           ;REPORT ERROR
7531 050662 000240          NOP
7532 050664 113760 001224 000010 70$:  MOV     PORTA,RMCS2(RO) ;SELECT PORT A
7533 050672 013737 001224 001240   MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7534
7535 ;*****
7536 ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A
7537
7538 050700 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
7539 050704 016037 000012 001126   MOV     RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
7540 050712 012737 000012 001122   MOV     #RMDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7541 050720 060037 001122          ADD     RO,$B0ADR    ;ADD RHI1 BASE ADDRESS
7542 050724 005037 001124          CLR     $GDDAT       ;WHAT REGISTER SHOULD BE
7543 050730 013737 001126 001164   MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7544 050736 042737 077777 001164   BIC     #1CATA,$TMP0 ;SAVE SPECIFIED BITS
7545 050744 023737 001124 001164   CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
7546 050752 001414          BEQ     71$          ;BR IF OK
7547 050754 013737 001126 001174   MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
7548 050762 042737 100000 001174   BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
7549 050770 053737 001174 001124   BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7550 050776 104032          ERROR    32         ;TYPE MESSAGE 32
7551 051000 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
7552 051004 000240          NOP
7553
7554 ;*****
7555 051006 000004          SCOPE              ;LOOP ?
7556
7557

```

7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598 051010
7599 051010 005737 001300
7600 051014 001406
7601 051016 100002
7602 051020 000137 002676
7603 051024 012737 177777 001300
7604 051032 112737 000040 001102
7605 051040 012737 051062 001106
7606 051046 012737 051062 001110
7607 051054 012737 000031 001176
7608 051062 012706 001100
7609
7610
7611
7612
7613

```

*****
*
*VERIFY THAT A CHANGE IN UNIT READY SETS THE ATTENTION
*   FOR BOTH PORTS.
*
*THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.
*
*****
*
*****
*VERIFY THAT ATTENTION SETS WHEN THE DRIVE SWITCHES AFTER
*BEING RELEASED.
*
*THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"
*
*****
*
*****
*TEST 40   PORT 'A' SET VOLUME VALID TEST
*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
*
* A.   WITH PORT 'A' SELECTED, RESET AND SET "UNIT READY"
*       STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
*       IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
*       ATTENTION IS SET.
*
* B.   ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
*       COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
*       VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
*       IS SET.
*
* C.   RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR
*       PORT 'B'. VERIFY THAT ATTENTION IS STIL SET AND THAT
*       VOLUME VALID IS STIL RESET.
*
* D.   ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
*       PORT 'B' THEN RELEASE PORT 'B'.
*
*****
TST40:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      25         ;BR IF NOT
      BPL      15         ;BR IF JUST ENTERED TEST
      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
15:   MOV      #-1, KYBCTL ;SET SINGLE TEST INDICATOR
25:   MOV      #40, $STNM  ;TEST NUMBER
      MOV      #TEST40, $LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST40, $LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #25, $TIMES  ;DO 25. ITERATIONS
TEST40: MOV      #STACK, SP ;LOAD THE STACK POINTER
*****
*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.

```

```

7614
7615
7616
7617 051066 113760 001224 000010      MOV#B  PORTA, RMCS2(RO) ; SELECT PORT A
7618 051074 013737 001224 001242      MOV  PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
7619 051102 012760 000001 000024      MOV  ##DMD, RMMR1(RO) ; WRITE #DMD INTO RMMR1
7620 051110 013737 001226 001244      MOV  PORTB, OPPRT ; 'OPPOSITE' PORT ADDRESS
7621 051116 012760 001001 000024      MOV  #DMD!MUR, RMMR1(RO) ; SET UNIT READY
7622 051124 012760 000000 000024      MOV  #0, RMMR1(RO) ; RESET DIAGNOSTIC MODE
7623
7624
7625
7626 051132 005037 001250
7627 051136 016037 000012 001126
7628 051144 012737 000012 001122
7629 051152 060037 001122
7630 051156 012737 100000 001124
7631 051164 013737 001126 001164
7632 051172 042737 077777 001164
7633 051200 023737 001124 001164
7634 051206 001414
7635 051210 013737 001126 001174
7636 051216 042737 100000 001174
7637 051224 053737 001174 001124
7638 051232 104064
7639 051234 005137 001250
7640 051240 000240
7641 051242 005037 001250
7642 051246 016037 000012 001126
7643 051254 012737 000012 001122
7644 051262 060037 001122
7645 051266 005037 001124
7646 051272 013737 001126 001164
7647 051300 042737 177677 001164
7648 051306 023737 001124 001164
7649 051314 001414
7650 051316 013737 001126 001174
7651 051324 042737 000100 001174
7652 051332 053737 001174 001124
7653 051340 104065
7654 051342 005137 001250
7655 051346 000240
7656
7657
7658 051350 012760 000011 000000
7659 051356 012760 000021 000000
7660
7661
7662 051364 005037 001250
7663 051370 016037 000012 001126
7664 051376 012737 000012 001122
7665 051404 060037 001122
7666 051410 012737 000100 001124
7667 051416 013737 001126 001164
7668 051424 042737 177677 001164
7669 051432 023737 001124 001164

```

; SEIZE THE DRIVE THROUGH PORT A

```

;*****
; VERIFY THAT ATA=1, VV=0 FOR PORT A
CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
MOV  #RMDS1, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD  RO, $B0ADR ; ADD RH11 BASE ADDRESS
MOV  #ATA, $GDDAT ; WHAT REGISTER SHOULD BE
MOV  $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
BIC  #+CATA, $TMP0 ; SAVE SPECIFIED BITS
CMP  $GDDAT, $TMP0 ; COMPARE THE BITS
BEQ  66$ ; BR IF OK
MOV  $BDDAT, $TMP4 ; COPY 'BAD DATA'
BIC  #ATA, $TMP4 ; CLEAR THE MASKED BITS
BIS  $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 64 ; TYPE MESSAGE 64
COM  CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
66$:
NOP
CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
MOV  #RMDS1, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD  RO, $B0ADR ; ADD RH11 BASE ADDRESS
CLR  $GDDAT ; WHAT REGISTER SHOULD BE
MOV  $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
BIC  #+CVV, $TMP0 ; SAVE SPECIFIED BITS
CMP  $GDDAT, $TMP0 ; COMPARE THE BITS
BEQ  68$ ; BR IF OK
MOV  $BDDAT, $TMP4 ; COPY 'BAD DATA'
BIC  #VV, $TMP4 ; CLEAR THE MASKED BITS
BIS  $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 65 ; TYPE MESSAGE 65
COM  CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
68$:
NOP

```

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

```

MOV  #11, RMCS1(RO) ; DRIVE CLEAR
MOV  #21, RMCS1(RO) ; READ IN PRESET

```

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

```

CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
MOV  #RMDS1, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD  RO, $B0ADR ; ADD RH11 BASE ADDRESS
MOV  #VV, $GDDAT ; WHAT REGISTER SHOULD BE
MOV  $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
BIC  #+CVV, $TMP0 ; SAVE SPECIFIED BITS
CMP  $GDDAT, $TMP0 ; COMPARE THE BITS

```

```

7670 051440 001414          BEQ      705          ;BR IF OK
7671 051442 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
7672 051450 042737 000100 001174  BIC     #VV,$TMP4    ;CLEAR THE MASKED BITS
7673 051456 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7674 051464 104013          ERROR   13          ;TYPE MESSAGE 13
7675 051466 005137 001250          COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
7676 051472 000240          705:  NOP
7677 051474 005037 001250          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
7678 051500 016037 000012 001126  MOV     RMD$1(RO),$BDDAT ;GET CONTENTS OF RMD$1
7679 051506 012737 000012 001122  MOV     #RMD$1,$BDDAR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7680 051514 060037 001122          ADD     RO,$BDDAR   ;ADD RHI1 BASE ADDRESS
7681 051520 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
7682 051526 013737 001126 001164  MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
7683 051534 042737 024007 001164  BIC     #1C153770,$TMP0 ;SAVE SPECIFIED BITS
7684 051542 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
7685 051550 001414          BEQ     725          ;BR IF OK
7686 051552 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
7687 051560 042737 153770 001174  BIC     #153770,$TMP4 ;CLEAR THE MASKED BITS
7688 051566 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7689 051574 104010          ERROR   10          ;REPORT THE ERROR
7690 051576 005137 001250          COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
7691 051602 000240          725:  NOP
7692
7693 ;RELEASE PORT A AND SELECT PORT B
7694 ;VERIFY THAT ATA=1 AND VV=0 FOR PORT B
7695
7696 ;RELEASE THE DRIVE FROM PORT A
7697
7698 051604 113760 001224 000010  MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
7699 051612 013737 001224 001240  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7700 051620 012760 000013 000000  MOV     #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7701
7702 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7703
7704 051626 005037 001254          CLR     RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
7705 051632 012737 000012 001122  MOV     #RMD$1,$BDDAR ;FORM THE ADDRESS OF RMD$1 FOR TYPEOUT
7706 051640 060037 001122          ADD     RO,$BDDAR   ;ADD THE I/O BASE ADDRESS
7707 051644 012737 011600 001124  MOV     #MOL!PGM!DPR!DRY!$GDDAT ;COMPARISON CONSTANT
7708 051652 113760 001224 000010  MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.
7709 051660 016037 000012 001170  MOV     RMD$1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7710 051666 042737 024001 001170  BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7711 051674 013737 001170 001164  MOV     $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
7712 051702 042737 100100 001164  BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7713 051710 113760 001226 000010  MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B.
7714 051716 016037 000012 001172  MOV     RMD$1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7715 051724 042737 024001 001172  BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7716 051732 013737 001172 001166  MOV     $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
7717 051740 042737 100100 001166  BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7718 051746 023737 001164 001166  CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7719 051754 001006          BNE     745          ;BR IF NOT
7720 051756 005737 001164          TST     $TMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7721 051762 001037          BNE     765          ;BR IF NOT
7722 051764 104046          ERROR   46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7723 051766 000137 052152          JMP     785          ;BYPASS THE REST OF THE CHECKS
7724 051772 013737 001170 001126  745:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7725 052000 013737 001226 001240  MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```


PORT 'A' SET VOLUME VALID TEST

7726	052006	113760	001226	000010		MOVB	PORTB, RMCS2(RO)	; SELECT PORT B.
7727	052014	005737	001164			TST	\$TMPO	; SEE IF STATUS EQ 0 FROM PORT A.
7728	052020	001414				BEQ	75\$; BR IF ZERO
7729	052022	013737	001224	001240		MOV	PORTA, PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7730	052030	013737	001124	001126		MOV	\$TMP3, \$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
7731	052036	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	; SELECT PORT A.
7732	052044	005737	001166			TST	\$TMP1	; SEE IF STATUS EQ ZERO FROM PORT B.
7733	052050	001004				BNE	76\$; BR IF NOT
7734	052052	012737	177777	001254	75\$:	MOV	#-1, RELERR	; SET 'RELEASE ERROR' INDICATOR
7735	052060	104022				ERROR	22	; TYPE ERROR MESSAGE 22
7736	052062	013737	001170	001126	76\$:	MOV	\$TMP2, \$BDDAT	; LOOK FOR BIT FAILURES WHEN RMD51 READ
7737	052070	013737	001224	001240		MOV	PORTA, PTNBR	; CHANGE PORT NUMBER
7738	052076	042737	100100	001126		BIC	#ATA!VV, \$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
7739	052104	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; ALL BITS OK ?
7740	052112	001401				BEQ	77\$; BR IF OK FROM PORT A.
7741	052114	104007				ERROR	7	; REPORT ERROR
7742	052116	013737	001172	001126	77\$:	MOV	\$TMP3, \$BDDAT	; CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
7743	052124	013737	001226	001240		MOV	PORTB, PTNBR	; CHANGE PORT NUMBER
7744	052132	042737	100100	001126		BIC	#ATA!VV, \$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
7745	052140	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; SEE IF READ OK FROM PORT B.
7746	052146	001401				BEQ	78\$; BR IF OK
7747	052150	104007				ERROR	7	; REPORT ERROR
7748	052152	000240			78\$:	NOP		
7749	052154	113760	001226	000010		MOVB	PORTB, RMCS2(RO)	; SELECT PORT B
7750	052162	013737	001226	001240		MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7751	052170	005037	001250			CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
7752	052174	016037	000012	001126		MOV	RMD51(RO), \$BDDAT	; GET CONTENTS OF RMD51
7753	052202	012737	000012	001122		MOV	#RMD51, \$BDDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7754	052210	060037	001122			ADD	RO, \$BDDADR	; ADD RMI1 BASE ADDRESS
7755	052214	012737	100000	001124		MOV	#ATA, \$GDDAT	; WHAT REGISTER SHOULD BE
7756	052222	013737	001126	001164		MOV	\$BDDAT, \$TMPO	; MOVE REGISTER CONTENTS TO 'STMPO'
7757	052230	042737	077777	001164		BIC	#ICATA, \$TMPO	; SAVE SPECIFIED BITS
7758	052236	023737	001124	001164		CMP	\$GDDAT, \$TMPO	; COMPARE THE BITS
7759	052244	001414				BEQ	79\$; BR IF OK
7760	052246	013737	001126	001174		MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
7761	052254	042737	100000	001174		BIC	#ITA, \$TMP4	; CLEAR THE MASKED BITS
7762	052262	053737	001174	001124		BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7763	052270	104064				ERROR	64	; TYPE MESSAGE 64
7764	052272	005137	001250			COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
7765	052276	000240			79\$:	NOP		
7766	052300	005037	001250			CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
7767	052304	016037	000012	001126		MOV	RMD51(RO), \$BDDAT	; GET CONTENTS OF RMD51
7768	052312	012737	000012	001122		MOV	#RMD51, \$BDDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7769	052320	060037	001122			ADD	RO, \$BDDADR	; ADD RMI1 BASE ADDRESS
7770	052324	005037	001124			CLR	\$GDDAT	; WHAT REGISTER SHOULD BE
7771	052330	013737	001126	001164		MOV	\$BDDAT, \$TMPO	; MOVE REGISTER CONTENTS TO 'STMPO'
7772	052336	042737	177677	001164		BIC	#ICVV, \$TMPO	; SAVE SPECIFIED BITS
7773	052344	023737	001124	001164		CMP	\$GDDAT, \$TMPO	; COMPARE THE BITS
7774	052352	001414				BEQ	81\$; BR IF OK
7775	052354	013737	001126	001174		MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
7776	052362	042737	000100	001174		BIC	#VV, \$TMP4	; CLEAR THE MASKED BITS
7777	052370	053737	001174	001124		BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7778	052376	104065				ERROR	65	; TYPE MESSAGE 65
7779	052400	005137	001250			COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
7780	052404	000240			81\$:	NOP		
7781								

```

7782 ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B,
7783 ;THEN RELEASE PORT B
7784 052406 012760 000011 000000 MOV #11,RMCS1(RO) ;DRIVE CLEAR
7785 052414 012760 000021 000000 MOV #21,RMCS1(RO) ;READ IN PRESET
7786
7787 ;RELEASE THE DRIVE FROM PORT B
7788
7789 052422 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
7790 052430 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7791 052436 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
7792
7793 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7794
7795 052444 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7796 052450 012737 000012 001122 MOV #RMDS1,$BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7797 052456 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
7798 052462 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
7799 052470 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
7800 052476 016037 000012 001170 MOV RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7801 052504 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7802 052512 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'STMP0'
7803 052520 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7804 052526 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
7805 052534 016037 000012 001172 MOV RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7806 052542 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7807 052550 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'STMP1'
7808 052556 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7809 052564 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7810 052572 001006 BNE 83$ ;BR IF NOT
7811 052574 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7812 052600 001037 BNE 85$ ;BR IF NOT
7813 052602 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7814 052604 000137 052770 JMP 87$ ;BYPASS THE REST OF THE CHECKS
7815 052610 013737 001170 001126 83$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7816 052616 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7817 052624 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
7818 052632 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7819 052636 001414 BEQ 84$ ;BR IF ZERO
7820 052640 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7821 052646 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7822 052654 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
7823 052662 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7824 052666 001004 BNE 85$ ;BR IF NOT
7825 052670 012737 177777 001254 84$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7826 052676 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
7827 052700 013737 001170 001126 85$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7828 052706 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7829 052714 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
7830 052722 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
7831 052730 001401 BEQ 86$ ;BR IF OK FROM PORT A.
7832 052732 104007 ERROR 7 ;REPORT ERROR
7833 052734 013737 001172 001126 86$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7834 052742 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
7835 052750 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
7836 052756 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
7837 052764 001401 BEQ 87$ ;BR IF OK
    
```

7838 052766 104007
7839 052770 000240
7840
7841 052772 000004
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864

ERROR 7 ;REPORT ERROR
87\$: NOP
50\$: SCOPE

```
*****
*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 STIL SET AND THAT
* VOLUME VALID IS STIL RESET.
*
* D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
* PORT 'A' THEN RELEASE PORT 'A'.
*****
```

7865 052774
7866 052774 005737 001300
7867 053000 001406
7868 053002 100002
7869 053004 000137 002676
7870 053010 012737 177777 001300
7871 053016 112737 000041 001102
7872 053024 (2737 053046 001106
7873 053032 012737 053046 001110
7874 053040 012737 000031 001176
7875 053046 012706 001100
7876
7877
7878
7879
7880
7881
7882
7883

```
*****
*ST41:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #41,$STSTNM ;TEST NUMBER
MOV #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25. ITERATIONS
TEST41: MOV #STACK,$SP ;LOAD THE STACK POINTER
*****
```

7884 053052 113760 001226 000010
7885 053060 013737 001226 001242
7886 053066 012760 000001 000024
7887 053074 013737 001224 001244
7888 053102 012760 001001 000024
7889 053110 012760 000000 000024
7890
7891
7892
7893 053116 005037 001250

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

```
*****
;VERIFY THAT ATA=1,VV=0 FOR PORT B
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
*****
```

7894	053122	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
7895	053130	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7896	053136	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
7897	053142	012737	100000	001124	MOV	#ATA, \$GDDAT	; WHAT REGISTER SHOULD BE
7898	053150	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'
7899	053156	042737	077777	001164	BIC	#+CATA, \$TMP0	; SAVE SPECIFIED BITS
7900	053164	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
7901	053172	001414			BEQ	66\$; BR IF OK
7902	053174	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
7903	053202	042737	100000	001174	BIC	#ATA, \$TMP4	; CLEAR THE MASKED BITS
7904	053210	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7905	053216	104064			ERROR	64	; TYPE MESSAGE 64
7906	053220	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
7907	053224	000240			66\$: NOP		
7908	053226	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
7909	053232	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
7910	053240	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7911	053246	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
7912	053252	005037	001124		CLR	\$GDDAT	; WHAT REGISTER SHOULD BE
7913	053256	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'
7914	053264	042737	177677	001164	BIC	#+CVV, \$TMP0	; SAVE SPECIFIED BITS
7915	053272	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
7916	053300	001414			BEQ	68\$; BR IF OK
7917	053302	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
7918	053310	042737	000100	001174	BIC	#VV, \$TMP4	; CLEAR THE MASKED BITS
7919	053316	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7920	053324	104065			ERROR	65	; TYPE MESSAGE 65
7921	053326	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
7922	053332	000240			68\$: NOP		
7923							
7924							; ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B
7925	053334	012760	000011	000000	MOV	#11, RMCS1(RO)	; DRIVE CLEAR
7926	053342	012760	000021	000000	MOV	#21, RMCS1(RO)	; READ IN PRESET
7927							
7928							; VERIFY ATA=0 AND VV=1 FOR PORT B
7929	053350	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
7930	053354	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
7931	053362	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7932	053370	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
7933	053374	012737	000100	001124	MOV	#VV, \$GDDAT	; WHAT REGISTER SHOULD BE
7934	053402	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'
7935	053410	042737	177677	001164	BIC	#+CVV, \$TMP0	; SAVE SPECIFIED BITS
7936	053416	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
7937	053424	001414			BEQ	70\$; BR IF OK
7938	053426	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
7939	053434	042737	000100	001174	BIC	#VV, \$TMP4	; CLEAR THE MASKED BITS
7940	053442	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7941	053450	104013			ERROR	13	; TYPE MESSAGE 13
7942	053452	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
7943	053456	000240			70\$: NOP		
7944	053460	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
7945	053464	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
7946	053472	012737	000012	001122	MOV	#RMDS1, \$BDAOR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
7947	053500	060037	001122		ADD	RO, \$BDAOR	; ADD RH11 BASE ADDRESS
7948	053504	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	; WHAT REGISTER SHOULD BE
7949	053512	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'

```

7950 053520 042737 0240U7 001164      BIC      #1C153770, $TMP0      ;SAVE SPECIFIED BITS
7951 053526 023737 001124 001164      CMP      $GDDAT, $TMP0      ;COMPARE THE BITS
7952 053534 001414                      BEQ      72$                 ;BR IF OK
7953 053536 013737 001126 001174      MOV      $BDDAT, $TMP4      ;COPY 'BAD DATA'
7954 053544 042737 153770 001174      BIC      #153770, $TMP4      ;CLEAR THE MASKED BITS
7955 053552 053737 001174 001124      BIS      $TMP4, $GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
7956 053560 104010                      ERROR    10                 ;REPORT THE ERROR
7957 053562 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
7958 053566 000240                      72$:  NOP
7959
7960                      ;RELEASE PORT B AND SELECT PORT A
7961                      ;VERIFY THAT ATA=1 AND VV=0 FOR PORT A
7962
7963                      ;RELEASE THE DRIVE FROM PORT B
7964
7965 053570 113760 001226 000010      MOV      PORTB, RMCS2(RO)    ;SELECT PORT B
7966 053576 013737 001226 001240      MOV      PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7967 053604 012760 000013 000000      MOV      #13, RMCS1(RO)    ;ISSUE RELEASE THROUGH PORT B
7968
7969                      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7970
7971 053612 005037 001254                      CLR      RELERR            ;CLEAR THE 'RELEASE ERROR' INDICATOR
7972 053616 012737 000012 001122      MOV      #RMS1, $BDAOR     ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
7973 053624 060037 001122                      ADD      RO, $BDAOR        ;ADD THE I/O BASE ADDRESS
7974 053630 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
7975 053636 113760 001224 000010      MOV      PORTA, RMCS2(RO)  ;SELECT PORT A.
7976 053644 016037 000012 001170      MOV      RMS1(RO), $TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7977 053652 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
7978 053660 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
7979 053666 042737 100100 001164      BIC      #ATA!VV, $TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7980 053674 113760 001226 000010      MOV      PORTB, RMCS2(RO)  ;SELECT PORT B.
7981 053702 016037 000012 001172      MOV      RMS1(RO), $TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7982 053710 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
7983 053716 013737 001172 001166      MOV      $TMP3, $TMP1      ;COPY IT INTO '$TMP1'
7984 053724 042737 100100 001166      BIC      #ATA!VV, $TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7985 053732 023737 001164 001166      CMP      $TMP0, $TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7986 053740 001006                      BNE      74$               ;BR IF NOT
7987 053742 005737 001164                      TST      $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7988 053746 001037                      BNE      76$               ;BR IF NOT
7989 053750 104046                      ERROR    46                 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7990 053752 000137 054136                      JMP      78$               ;BYPASS THE REST OF THE CHECKS
7991 053756 013737 001170 001126 74$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7992 053764 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7993 053772 113760 001226 000010      MOV      PORTB, RMCS2(RO)  ;SELECT PORT B.
7994 054000 005737 001164                      TST      $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
7995 054004 001414                      BEQ      75$               ;BR IF ZERO
7996 054006 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7997 054014 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
7998 054022 113760 001224 000010      MOV      PORTA, RMCS2(RO)  ;SELECT PORT A.
7999 054030 005737 001166                      TST      $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
8000 054034 001004                      BNE      76$               ;BR IF NOT
8001 054036 012737 177777 001254 75$:  MOV      #-1, RELERR       ;SET 'RELEASE ERROR' INDICATOR
8002 054044 104022                      ERROR    22                 ;TYPE ERROR MESSAGE 22
8003 054046 013737 001170 001126 76$:  MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8004 054054 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT NUMBER
8005 054062 042737 100100 001126      BIC      #ATA!VV, $BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT

```

```

8006 054070 023737 001124 001126    CMP    $GDDAT,$BDDAT    ;ALL BITS OK ?
8007 054076 001401                    BEQ    77$              ;BR IF OK FROM PORT A.
8008 054100 104007                    ERROR  7               ;REPORT ERROR
8009 054102 013737 001172 001126 77$:  MOV    $TMP3,$BDDAT    ;CHECK RMD$1 FOR BIT FAILURES - FROM PORT B.
8010 054110 013737 001226 001240    MOV    PORTB,PTNBR    ;CHANGE PORT NUMBER
8011 054116 042737 100100 001126    BIC    #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
8012 054124 023737 001124 001126    CMP    $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8013 054132 001401                    BEQ    78$              ;BR IF OK
8014 054134 104007                    ERROR  7               ;REPORT ERROR
8015 054136 000240                    NOP
8016 054140 113760 001224 000010 78$:  MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
8017 054146 013737 001224 001240    MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8018 054154 005037 001250                    CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8019 054160 016037 000012 001126    MOV    RMD$1(RO), $BDDAT ;GET CONTENTS OF RMD$1
8020 054166 012737 000012 001122    MOV    #RMD$1,$B0ADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8021 054174 060037 001122                    ADD    RO,$B0ADR     ;ADD RHI1 BASE ADDRESS
8022 054200 012737 100000 001124    MOV    #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
8023 054206 013737 001126 001164    MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO 'TMP0'
8024 054214 042737 077777 001164    BIC    #+CATA,$TMP0  ;SAVE SPECIFIED BITS
8025 054222 023737 001124 001164    CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
8026 054230 001414                    BEQ    79$              ;BR IF OK
8027 054232 013737 001126 001174    MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
8028 054240 042737 100000 001174    BIC    #ATA,$TMP4   ;CLEAR THE MASKED BITS
8029 054246 053737 001174 001124    BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
8030 054254 104064                    ERROR  64              ;TYPE MESSAGE 64
8031 054256 005137 001250                    COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8032 054262 000240                    NOP
8033 054264 005037 001250                    CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8034 054270 016037 000012 001126    MOV    RMD$1(RO), $BDDAT ;GET CONTENTS OF RMD$1
8035 054276 012737 000012 001122    MOV    #RMD$1,$B0ADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8036 054304 060037 001122                    ADD    RO,$B0ADR     ;ADD RHI1 BASE ADDRESS
8037 054310 005037 001124                    CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
8038 054314 013737 001126 001164    MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO 'TMP0'
8039 054322 042737 177677 001164    BIC    #+CVV,$TMP0  ;SAVE SPECIFIED BITS
8040 054330 023737 001124 001164    CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
8041 054336 001414                    BEQ    81$              ;BR IF OK
8042 054340 013737 001126 001174    MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
8043 054346 042737 000100 001174    BIC    #VV,$TMP4   ;CLEAR THE MASKED BITS
8044 054354 053737 001174 001124    BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
8045 054362 104065                    ERROR  65              ;TYPE MESSAGE 65
8046 054364 005137 001250                    COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8047 054370 000240                    NOP
8048
8049 ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A,
8050 ;THEN RELEASE PORT A
8051 054372 012760 000011 000000    MOV    #11,RMCS1(RO) ;DRIVE CLEAR
8052 054400 012760 000021 000000    MOV    #21,RMCS1(RO) ;READ IN PRESET
8053
8054 ;RELEASE THE DRIVE FROM PORT A
8055
8056 054406 113760 001224 000010    MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
8057 054414 013737 001224 001240    MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8058 054422 012760 000013 000000    MOV    #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
8059
8060 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8061

```

```

8062 054430 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8063 054434 012737 000012 001122 MOV #RMS1,$BDAOR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
8064 054442 060037 001122 ADD RD,$BDAOR ;ADD THE I/O BASE ADDRESS
8065 054446 012737 011600 001124 MOV #MOL!PGH!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
8066 054454 113760 001224 000010 MOVB PORTA,RMCS2(RD) ;SELECT PORT A.
8067 054462 016037 000012 001170 MOV RMS1(RD),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8068 054470 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DON' CARES
8069 054476 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
8070 054504 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8071 054512 113760 001226 000010 MOVB PORTB,RMCS2(RD) ;SELECT PORT B.
8072 054520 016037 000012 001172 MOV RMS1(RD),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8073 054526 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
8074 054534 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
8075 054542 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8076 054550 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8077 054556 001006 BNE 83$ ;BR IF NOT
8078 054560 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8079 054564 001037 BNE 85$ ;BR IF NOT
8080 054566 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8081 054570 000137 054754 JMP 87$ ;BYPASS THE REST OF THE CHECKS
8082 054574 013737 001170 001126 83$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8083 054602 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8084 054610 113760 001226 000010 MOVB PORTB,RMCS2(RD) ;SELECT PORT B.
8085 054616 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8086 054622 001414 BEQ 84$ ;BR IF ZERO
8087 054624 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8088 054632 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8089 054640 113760 001224 000010 MOVB PORTA,RMCS2(RD) ;SELECT PORT A.
8090 054646 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8091 054652 001004 BNE 85$ ;BR IF NOT
8092 054654 012737 177777 001254 84$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8093 054662 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
8094 054664 013737 001170 001126 85$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8095 054672 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8096 054700 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
8097 054706 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
8098 054714 001401 BEQ 86$ ;BR IF OK FROM PORT A.
8099 054716 104007 ERROR 7 ;REPORT ERROR
8100 054720 013737 001172 001126 86$: MOV $TMP3,$BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
8101 054726 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8102 054734 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
8103 054742 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8104 054750 001401 BEQ 87$ ;BR IF OK
8105 054752 104007 ERROR 7 ;REPORT ERROR
8106 054754 000240 87$: NOP
8107
8108 054756 000004 50$: SCOPE
8109
8110
8111
8112
8113
8114
8115
8116
8117

```

```

*****
*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 RMS1.
*

```

```

8118
8119
8120
8121
8122
8123
8124
8125 054760
8126 054760 005737 001300
8127 054764 001406
8128 054766 100002
8129 054770 000137 002676
8130 054774 012737 177777 001300
8131 055002 112737 000042 001102
8132 055010 012737 055032 001106
8133 055016 012737 055032 001110
8134 055024 012737 000002 001176
8135 055032 012706 001100
8136
8137
8138
8139 055036 113760 001224 000010
8140 055044 005060 000012
8141 055050 012760 000011 000000
8142 055056 012760 000013 000000
8143 055064 113760 001226 000010
8144 055072 005060 000012
8145 055076 012760 000011 000000
8146 055104 012760 000013 000000
8147
8148
8149
8150
8151 055112 113760 001224 000010
8152 055120 013737 001224 001242
8153 055126 005060 000012
8154 055132 013737 001226 001244
8155
8156
8157
8158
8159 055140 012760 177777 000014
8160
8161
8162
8163
8164 055146 005037 001256
8165 055152 012737 003720 001260
8166 055160 113760 001226 000010
8167 055166 013737 001226 001240
8168
8169
8170
8171
8172 055174 005760 000012
8173 055200 001004

```

```

;* 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.
;*****
TST42:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOVB     #42,$STSTNM ;TEST NUMBER
        MOV      #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
        MOV      #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
        MOV      #2,$TIMES      ;DO 2. ITERATIONS
TEST42:  MOV      #STACK,$SP    ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
        MOVB     PORTA,$RMCS2($R0) ;SELECT PORT #A
        CLR      $RMS1($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      $RMS1($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
        MOVB     PORTA,$RMCS2($R0) ;SELECT PORT A
        MOV      PORTA,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
        CLR      $RMS1($R0)       ;WRITE RMS1
        MOV      PORTB,$OPPRT     ;'OPPOSITE' PORT ADDRESS
;*****
;FORCE AN ATTENTION BY SETTING ERRORS.
        MOV      #-1,$RMER1($R0) ;SET ERROR BITS
;*****
;START THE TIMER
        CLR      TIME              ;CLEAR THE ELAPSED TIME COUNTER
        MOV      #2000,$WATCH      ;SET WATCH TO 2000 MS
        MOVB     PORTB,$RMCS2($R0) ;SELECT PORT B
        MOV      PORTB,$PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;*****
;WAIT FOR DRIVE TO TIMEOUT
1$:      TST      $RMS1($R0)      ;WAIT FOR THE DRIVE TO BE RELEASED
        BNE     2$              ;BR IF DRIVE RELEASED

```


8174 055202 005737 001260
8175 055206 001372
8176 055210 104036
8177 055212
8178 055212 113760 001224 000010
8179 055220 013737 001224 001240
8180
8181
8182
8183
8184 055226 005037 001250
8185 055232 016037 000012 001126
8186 055240 012737 000012 001122
8187 055246 060037 001122
8188 055252 012737 040000 001124
8189 055260 013737 001126 001164
8190 055266 042737 137777 001164
8191 055274 023737 001124 001164
8192 055302 001414
8193 055304 013737 001126 001174
8194 055312 042737 040000 001174
8195 055320 053737 001174 001124
8196 055326 104023
8197 055330 005137 001250
8198 055334 000240
8199
8200
8201
8202
8203 055336 005037 001250
8204 055342 016037 000014 001126
8205 055350 012737 000014 001122
8206 055356 060037 001122
8207 055362 012737 177777 001124
8208 055370 023737 001124 001126
8209 055376 001403
8210 055400 104010
8211 055402 005137 001250
8212 055406 000240
8213
8214
8215
8216
8217 055410 005037 001250
8218 055414 016037 000012 001126
8219 055422 012737 000012 001122
8220 055430 060037 001122
8221 055434 012737 100000 001124
8222 055442 013737 001126 001164
8223 055450 042737 077777 001164
8224 055456 023737 001124 001164
8225 055464 001414
8226 055466 013737 001126 001174
8227 055474 042737 100000 001174
8228 055502 053737 001174 001124
8229 055510 104041

```

TST WATCH ;WATCH AT ZERO ?
BNE 15 ;BR IF NOT
ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS

2$:
MOV PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;THE ERROR BIT ('ERR') IN RMDS1 SHOULD STILL BE SET

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #+CERR, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ERR, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 23 ;TYPE MESSAGE 23
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

66$:
NOP

;*****
;THE ERROR REGISTER SHOULD CONTAIN 1'S

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
MOV #RMER1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
BEQ 68$ ;BR IF OK
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

68$:
NOP

;*****
;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #+CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 70$ ;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
ERROR 41 ;TYPE MESSAGE 41

```

```

8230 055512 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8231 055516 000240          70$:    NOP
8232
8233
8234                                     ;*****
8235                                     ;
8236                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8237
8238 055520 005037 001254          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8239 055524 012737 000012 001122    MOV      #RMS1,$B0ADR    ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
8240 055532 060037 001122          ADD      R0,$B0ADR      ;ADD THE I/O BASE ADDRESS
8241 055536 012737 051700 001124    MOV      #51700,$G00AT  ;COMPARISON CONSTANT
8242 055544 113760 001224 000010    MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
8243 055552 016037 000012 001170    MOV      RMS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8244 055560 042737 024001 001170    BIC     #PIP!WRL!0M,$TMP2 ;CLEAR DONT CARES
8245 055566 013737 001170 001164    MOV      $TMP2,$TMP0     ;COPY IT INTO 'TMP0'
8246 055574 042737 100100 001164    BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8247 055602 113760 001226 000010    MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
8248 055610 016037 000012 001172    MOV      RMS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8249 055616 042737 024001 001172    BIC     #PIP!WRL!0M,$TMP3 ;CLEAR DONT CARES
8250 055624 013737 001172 001166    MOV      $TMP3,$TMP1     ;COPY IT INTO 'TMP1'
8251 055632 042737 100100 001166    BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8252 055640 023737 001164 001166    CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8253 055646 001006          BNE     72$             ;BR IF NOT
8254 055650 005737 001164          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8255 055654 001045          BNE     74$             ;BR IF NOT
8256 055656 104046          ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8257 055660 000137 056060          JMP     76$             ;BYPASS THE REST OF THE CHECKS
8258 055664 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8259 055672 013737 001226 001240    MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8260 055700 113760 001226 000010    MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
8261 055706 005737 001164          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8262 055712 001414          BEQ     73$             ;BR IF ZERO
8263 055714 013737 001224 001240    MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8264 055722 013737 001172 001126    MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8265 055730 113760 001224 000010    MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
8266 055736 005737 001166          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8267 055742 001012          BNE     74$             ;BR IF NOT
8268 055744 012737 177777 001254 73$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8269 055752 012760 000011 000000    MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
8270 055760 012760 000013 000000    MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
8271 055766 104026          ERROR   26             ;TYPE ERROR MESSAGE 26
8272 055770 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8273 055776 013737 001224 001240    MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8274 056004 042737 100000 001126    BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
8275 056012 023737 001124 001126    CMP     $G00AT,$BDDAT  ;ALL BITS OK ?
8276 056020 001401          BEQ     75$             ;BR IF OK FROM PORT A.
8277 056022 104007          ERROR   7              ;REPORT ERROR
8278 056024 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT    ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
8279 056032 013737 001126 001240    MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8280 056040 042737 100000 001126    BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
8281 056046 023737 001124 001126    CMP     $G00AT,$BDDAT  ;SEE IF READ OK FROM PORT B.
8282 056054 001401          BEQ     76$             ;BR IF OK
8283 056056 104007          ERROR   7              ;REPORT ERROR
8284 056060 000240          76$:    NOP
8285

```

```

8286
8287
8288
8289 056062 113760 001226 000010      MOV  PORTB, RMCS2(RO) ;SELECT PORT B
8290 056070 013737 001226 001240      MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8291 056076 005037 001250      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8292 056102 016037 000012 001126      MOV  RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
8293 056110 012737 000012 001122      MOV  #RMDS1, $BOADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8294 056116 060037 001122      ADD  RO, $BOADR ;ADD RHI1 BASE ADDRESS
8295 056122 005037 001124      CLR  $GDDAT ;WHAT REGISTER SHOULD BE
8296 056126 013737 001126 001164      MOV  $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
8297 056134 042737 077777 001164      BIC  #1CATA, $TMP0 ;SAVE SPECIFIED BITS
8298 056142 023737 001124 001164      CMP  $GDDAT, $TMP0 ;COMPARE THE BITS
8299 056150 001414      BEQ  77$ ;BR IF OK
8300 056152 013737 001126 001174      MOV  $BDDAT, $TMP4 ;COPY 'BAD DATA'
8301 056160 042737 100000 001174      BIC  #ATA, $TMP4 ;CLEAR THE MASKED BITS
8302 056166 053737 001174 001124      BIS  $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
8303 056174 104052      ERROR 52 ;TYPE MESSAGE 52
8304 056176 005137 001250      COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8305 056202 000240      77$: NOP
8306
8307 ;CLEAR ATTENTION BIT FOR PORT A
8308
8309 056204 113760 001224 000010      MOV  PORTA, RMCS2(RO) ;SELECT PORT #A
8310 056212 005060 000012      CLR  RMDS1(RO) ;SEIZE THE DRIVE
8311 056216 012760 000011 000000      MOV  #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
8312 056224 012760 000013 000000      MOV  #13, RMCS1(RO) ;RELEASE THE DRIVE
8313 056232 000004      3$: SCOPE ;LOOP ?
8314
8315 ;*****
8316 ;*TEST 43 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
8317 ;*
8318 ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
8319 ;*
8320 ;* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
8321 ;*
8322 ;* B. WRITE 1'S INTO RMR1 THROUGH PORT 'B'.
8323 ;*
8324 ;* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
8325 ;* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
8326 ;* PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
8327 ;*
8328 ;*****
8329 †ST43:
8330 056234 005737 001300      TST  KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
8331 056240 001406      BEQ  2$ ;BR IF NOT
8332 056242 100002      BPL  1$ ;BR IF JUST ENTERED TEST
8333 056244 000137 002676      JMP  EXEC ;RETURN & GET NEXT TEST NUMBER
8334 056250 012737 177777 001300 1$: MOV  #-1, KYBCTL ;SET SINGLE TEST INDICATOR
8335 056256 112737 000043 001102 2$: MOV  #43, $STNM ;TEST NUMBER
8336 056264 012737 056306 001106      MOV  #TEST43, $LPADR ;LOAD LOOP ON TEST ADDRESS
8337 056272 012737 056306 001110      MOV  #TEST43, $LPERR ;LOAD LOOP ON ERROR ADDRESS
8338 056300 012737 000002 001176      MOV  #2, $TIMES ;DO 2. ITERATIONS
8339 056306 012706 001100      TEST43: MOV  #STACK, SP ;LOAD THE STACK POINTER
8340
8341 ;CLEAR ATTENTION BITS FOR BOTH PORTS

```

```

8342
8343 056312 113760 001224 000010      MOVB  PORTA,RMCS2(RO) ;SELECT PORT #A
8344 056320 005060 000012 000000      CLR   RMDS1(RO)      ;SEIZE THE DRIVE
8345 056324 012760 000011 000000      MOV   #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
8346 056332 012760 000013 000000      MOV   #13,RMCS1(RO) ;RELEASE THE DRIVE
8347 056340 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT #B
8348 056346 005060 000012 000000      CLR   RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
8349 056352 012760 000011 000000      MOV   #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
8350 056360 012760 000013 000000      MOV   #13,RMCS1(RO) ;RELEASE THE DRIVE
8351                                     ;*****
8352                                     ;SEIZE THE DRIVE THROUGH PORT B
8353
8354
8355 056366 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
8356 056374 013737 001226 001242      MOV   PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
8357 056402 005060 000012 000000      CLR   RMDS1(RO)      ;WRITE RMDS1
8358 056406 013737 001224 001244      MOV   PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
8359
8360                                     ;*****
8361                                     ;FORCE AN ATTENTION BY SETTING ERRORS.
8362
8363 056414 012760 177777 000014      MOV   #-1,RMER1(RO) ;SET ERROR BITS
8364
8365                                     ;*****
8366                                     ;START THE TIMER
8367
8368 056422 005037 001256 000000      CLR   TIME           ;CLEAR THE ELAPSED TIME COUNTER
8369 056426 012737 003720 001260      MOV   #2000,WATCH    ;SET WATCH TO 2000 MS
8370 056434 113760 001224 000010      MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
8371 056442 013737 001224 001240      MOV   PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8372
8373                                     ;*****
8374                                     ;WAIT FOR DRIVE TO TIMEOUT
8375
8376 056450 005760 000012 000000      1$:   TST   RMDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
8377 056454 001004 000000 000000      BNE   2$           ;BR IF DRIVE RELEASED
8378 056456 005737 001260 000000      TST   WATCH        ;WATCH AT ZERO ?
8379 056462 001372 000000 000000      BNE   1$           ;BR IF NOT
8380 056464 104036 000000 000000      ERROR 36           ;DRIVE NOT RELEASED WITHIN 2 SECONDS
8381 056466
8382 056466 113760 001226 000010      2$:   MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
8383 056474 013737 001226 001240      MOV   PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8384
8385                                     ;*****
8386                                     ;THE ERROR BIT ('ERR') IN RMDS1 SHOULD STILL BE SET
8387
8388 056502 005037 001250 000000      CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
8389 056506 016037 000012 001126      MOV   RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
8390 056514 012737 000012 001122      MOV   #RMDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8391 056522 060037 001122 000000      ADD   RO,SBADR      ;ADD RHI1 BASE ADDRESS
8392 056526 012737 040000 001124      MOV   #ERR,$GDDAT   ;WHAT REGISTER SHOULD BE
8393 056534 013737 001126 001164      MOV   SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
8394 056542 042737 137777 001164      BIC   #1CERR,$TMP0 ;SAVE SPECIFIED BITS
8395 056550 023737 001124 001164      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
8396 056556 001414 000000 000000      BEQ   66$           ;BR IF OK
8397 056560 013737 001126 001174      MOV   SBDDAT,$TMP4  ;COPY 'BAD DATA'

```

```

8398 056566 042737 040000 001174      BIC      #ERR,$TMP4      ;CLEAR THE MASKED BITS
8399 056574 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
8400 056602 104020 001250      ERROR   23             ;TYPE MESSAGE 23
8401 056604 005137 001250      COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8402 056610 000240 66$:      NOP
8403
8404      ;*****
8405      ;THE ERROR REGISTER SHOULD CONTAIN 1'S
8406
8407 056612 005037 001250      CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8408 056616 016037 000014 001126      MOV     RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
8409 056624 012737 000014 001122      MOV     #RMER1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8410 056632 060037 001122      ADD     RO,$B0ADR      ;ADD RH11 BASE ADDRESS
8411 056636 012737 177777 001124      MOV     #177777,$GDDAT ;WHAT REGISTER SHOULD BE
8412 056644 023737 001124 001126      CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8413 056652 001403 68$:      BEQ     68$           ;BR IF OK
8414 056654 104010 001250      ERROR   10             ;REPORT THE ERROR
8415 056656 005137 001250      COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8416 056662 000240 68$:      NOP
8417
8418      ;*****
8419      ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
8420
8421 056664 005037 001250      CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8422 056670 016037 000012 001126      MOV     RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
8423 056676 012737 000012 001122      MOV     #RMDS1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8424 056704 060037 001122      ADD     RO,$B0ADR      ;ADD RH11 BASE ADDRESS
8425 056710 012737 100000 001124      MOV     #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
8426 056716 013737 001126 001164      MOV     $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
8427 056724 042737 077777 001164      BIC     #CATA,$TMP0    ;SAVE SPECIFIED BITS
8428 056732 023737 001124 001164      CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
8429 056740 001414 70$:      BEQ     70$           ;BR IF OK
8430 056742 013737 001126 001174      MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
8431 056750 042737 100000 001174      BIC     #ATA,$TMP4     ;CLEAR THE MASKED BITS
8432 056756 053737 001174 001124      BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
8433 056764 104041 001250      ERROR   41             ;TYPE MESSAGE '1'
8434 056766 005137 001250      COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8435 056772 000240 70$:      NOP
8436
8437      ;*****
8438      ;
8439      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8440
8441
8442 056774 005037 001254      CLR     RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
8443 057000 012737 000012 001122      MOV     #RMDS1,$B0ADR   ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
8444 057006 060037 001122      ADD     RO,$B0ADR      ;ADD THE I/O BASE ADDRESS
8445 057012 012737 051700 001124      MOV     #51700,$GDDAT  ;COMPARISON CONSTANT
8446 057020 113760 001224 000010      MOV     PORTA,RMCS2(RO) ;SELECT PORT A.
8447 057026 016037 000012 001170      MOV     RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8448 057034 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
8449 057042 013737 001170 001164      MOV     $TMP2,$TMP0    ;COPY IT INTO 'TMP0'
8450 057050 042737 100100 001164      BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8451 057056 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B.
8452 057064 016037 000012 001172      MOV     RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8453 057072 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES

```

```

8454 057100 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO 'TMP1'
8455 057106 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8456 057114 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8457 057122 001006                BNE      72$            ;BR IF NOT
8458 057124 005737 001164                TST      $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8459 057130 001045                BNE      74$            ;BR IF NOT
8460 057132 104046                ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8461 057134 000137 057334                JMP      76$            ;BYPASS THE REST OF THE CHECKS
8462 057140 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8463 057146 013737 001226 001240      MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8464 057154 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B.
8465 057162 005737 001164                TST      $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
8466 057166 001414                BEQ      73$            ;BR IF ZERO
8467 057170 013737 001224 001240      MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8468 057176 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8469 057204 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A.
8470 057212 005737 001166                TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
8471 057216 001012                BNE      74$            ;BR IF NOT
8472 057220 012737 177777 001254 73$:  MOV      #-1,RELEERR    ;SET 'RELEASE ERROR' INDICATOR
8473 057226 012760 000011 000000      MOV      #11,RMCS1(RO)  ;CLEAR THE DRIVE
8474 057234 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
8475 057242 104026                ERROR   26             ;TYPE ERROR MESSAGE 26
8476 057244 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8477 057252 013737 001224 001240      MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
8478 057260 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
8479 057266 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;ALL BITS OK ?
8480 057274 001401                BEQ      75$            ;BR IF OK FROM PORT A.
8481 057276 104007                ERROR   7              ;REPORT ERROR
8482 057300 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT    ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8483 057306 013737 001226 001240      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
8484 057314 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
8485 057322 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;SEE IF READ OK FROM PORT B.
8486 057330 001401                BEQ      76$            ;BR IF OK
8487 057332 104007                ERROR   7              ;REPORT ERROR
8488 057334                NOP
8489
8490 ;:*****
8491 ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET
8492
8493 057336 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A
8494 057344 013737 001224 001240      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8495 057352 005037 001250                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8496 057356 016037 000012 001126      MOV      RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
8497 057364 012737 000012 001122      MOV      #RMDS1,$BDDAT? ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8498 057372 060037 001122                ADD      RO,$BDDADR     ;ADD RHI1 BASE ADDRESS
8499 057376 005037 001124                CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
8500 057402 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
8501 057410 042737 077777 001164      BIC      #ICATA,$TMP0   ;SAVE SPECIFIED BITS
8502 057416 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
8503 057424 001414                BEQ      77$            ;BR IF OK
8504 057426 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
8505 057434 042737 100000 001174      BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
8506 057442 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
8507 057450 104052                ERROR   52             ;TYPE MESSAGE 52
8508 057452 005137 001250                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8509 057456                NOP
77$:

```

```

8510
8511
8512
8513 057460 113760 001226 000010
8514 057466 005060 000012
8515 057472 012760 000011 000000
8516 057500 012760 000013 000000
8517 057506 000004

```

```

;CLEAR ATTENTION BIT FOR PORT B
MOV B PORTB, RMCS2(RO) ;SELECT PORT #B
CLR RMDS1(RO) ;SEIZE THE DRIVE
MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
3$: SCOPE ;LOOP ?

```

```

8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535

```

```

*****
*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 RMDS1.
*
* B. WAIT 500 MS AND READ RMDS1 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.
*****

```

```

8536 057510
8537 057510 005737 001300
8538 057514 001406
8539 057516 100002
8540 057520 000137 002676
8541 057524 012737 177777 001300
8542 057532 112737 000044 001102
8543 057540 012737 057562 001106
8544 057546 012737 057562 001110
8545 057554 012737 000002 001176
8546 057562 012706 001100
8547
8548
8549

```

```

*****
*ST44:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #44, $STNM ;TEST NUMBER
MOV #TEST44, $LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST44, $LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #2, $TIMES ;DO 2. ITERATIONS
TEST44: MOV #STACK, SP ;LOAD THE STACK POINTER

```

```

8550 057566 113760 001224 000010
8551 057574 005060 000012
8552 057600 012760 000011 000000
8553 057606 012760 000013 000000
8554 057614 113760 001226 000010
8555 057622 005060 000012
8556 057626 012760 000011 000000
8557 057634 012760 000013 000000
8558
8559
8560
8561
8562
8563 057642 113760 001224 000010
8564 057650 013737 001224 001242
8565 057656 005060 000012

```

```

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOV B PORTA, RMCS2(RO) ;SELECT PORT #A
CLR RMDS1(RO) ;SEIZE THE DRIVE
MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
MOV B PORTB, RMCS2(RO) ;SELECT PORT #B
CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(RO) ;RELEASE THE DRIVE

```

```

8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599

```

```

*****
;SEIZE THE DRIVE THROUGH PORT A
MOV B PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDS1(RO) ;WRITE RMDS1

```

```

8566 057662 013737 001226 001244      MOV      PORTB,OPPRT      ;'OPPOSITE' PORT ADDRESS
8567
8568      ;*****
8569      ;WAIT 500 MS
8570
8571
8572      ;*****
8573      ;START THE TIMER
8574
8575 057670 005037 001256      CLR      TIME            ;CLEAR THE ELAPSED TIME COUNTER
8576 057674 012737 000764 001260      MOV      #500.,WATCH    ;SET WATCH TO 500 MS
8577 057702 005737 001260      1$:     TST      WATCH      ;WATCH EQUAL TO ZERO
8578 057706 001375      BNE     1$              ;BR IF NOT
8579
8580      ;*****
8581      ;START THE TIMER
8582
8583 057710 005037 001256      CLR      TIME            ;CLEAR THE ELAPSED TIME COUNTER
8584 057714 012737 003720 001260      MOV      #2000.,WATCH   ;SET WATCH TO 2000 MS
8585
8586      ;*****
8587      ;RETRIGGER THE TIMEOUT ONE-SHOT
8588
8589 057722 005760 000012      TST     RMD51(RO)       ;RETRIGGER THE ONE-SHOT
8590 057726 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B
8591 057734 013737 001226 001240      MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8592 057742 005760 000012      2$:     TST     RMD51(RO) ;WAIT FOR TIMEOUT
8593 057746 001004      BNE     3$              ;BR IF TIMEOUT OCCURRED
8594 057750 005737 001260      TST     WATCH          ;WATCH EQUAL TO ZERO ?
8595 057754 001372      BNE     2$              ;BR IF NOT
8596 057756 104036      ERROR  36              ;NO TIMEOUT WITHIN 2 SECONDS
8597 057760 013737 001256 001276      3$:     MOV     TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
8598
8599      ;*****
8600
8601      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8602
8603 057766 005037 001254      CLR     RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
8604 057772 012737 000012 001122      MOV     #RMD51,$BDADR   ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
8605 060000 060037 001122      ADD     RO,$BDADR       ;ADD THE I/O BASE ADDRESS
8606 060004 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8607 060012 113760 001224 000010      MOV     PORTA,RMCS2(RO) ;SELECT PORT A.
8608 060020 016037 000012 001170      MOV     RMD51(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8609 060026 042737 024001 001170      BIC     #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
8610 060034 013737 001170 001164      MOV     STMP2,STMP0     ;COPY IT INTO 'STMP0'
8611 060042 042737 100100 001164      BIC     #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8612 060050 113760 001226 000010      MOV     PORTB,RMCS2(RO) ;SELECT PORT B.
8613 060056 016037 000012 001172      MOV     RMD51(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8614 060064 042737 024001 001172      BIC     #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
8615 060072 013737 001172 001166      MOV     STMP3,STMP1     ;COPY IT INTO 'STMP1'
8616 060100 042737 100100 001166      BIC     #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8617 060106 023737 001164 001166      CMP     STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8618 060114 001006      BNE     66$            ;BR IF NOT
8619 060116 005737 001164      TST     STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8620 060122 001045      BNE     68$            ;BR IF NOT
8621 060124 104046      ERROR  46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED

```



```

8622 060126 000137 060312          JMP      70$          ;BYPASS THE REST OF THE CHECKS
8623 060132 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8624 060140 013737 001226 001240      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8625 060146 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B.
8626 060154 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
8627 060160 001414          BEQ     67$          ;BR IF ZERO
8628 060162 013737 001224 001240      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8629 060170 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8630 060176 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.
8631 060204 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
8632 060210 001012          BNE     68$          ;BR IF NOT
8633 060212 012737 177777 001254 67$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
8634 060220 012760 000011 000000      MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
8635 060226 012760 000013 000000      MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
8636 060234 104022          ERROR   22          ;TYPE ERROR MESSAGE 22
8637 060236 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8638 060244 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
8639 060252 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
8640 060260 001401          BEQ     69$          ;BR IF OK FROM PORT A.
8641 060262 104007          ERROR   7           ;REPORT ERROR
8642 060264 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8643 060272 013737 001226 001240      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
8644 060300 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8645 060306 001401          BEQ     70$          ;BR IF OK
8646 060310 104007          ERROR   7           ;REPORT ERROR
8647 060312 000240          70$:  NOP

```

```

;*****
;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

```

8652 060314 023737 001276 001264      CMP     TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
8653 060322 003004          BGT     4$           ;BR IF GREATER
8654 060324 023737 001276 001266      CMP     TIMES,TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
8655 060332 002001          BGE     +4          ;BR IF NOT
8656 060334 104025          4$:  ERROR   25          ;REPORT THE ERROR
8657 060336 000004          SCOPE ;LOOP ?

```

```

;*****
;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 RMDS1.
;
; B. WAIT 500 MS AND WRITE 0'B INTO RMDS1 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.

```

```

8675 060340          TST45:
8676 060340 005737 001300      TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
8677 060344 001406          BEQ     2$         ;BR IF NOT

```

E13

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 160
 T45 PORT 'B' RETRIGGER BY DEMAND TEST

SEQ 0162

```

8678 060346 100002          BPL      1$          ;BR IF JUST ENTERED TEST
8679 060350 000137 002676    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
8680 060354 012737 177777 001300 1$:  MOV     #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
8681 060362 112737 000045 001102 2$:  MOVVB  #45,$TSTNM  ;TEST NUMBER
8682 060370 012737 060412 001106    MOV     #TEST45,$LPADR ;LOAD LOOP ON TEST ADDRESS
8683 060376 012737 060412 001110    MOV     #TEST45,$LPERR ;LOAD LOOP ON ERROR ADDRESS
8684 060404 012737 000002 001176    MOV     #2,$TIMES     ;DO 2. ITERATIONS
8685 060412 012706 001100    TEST45: MOV    #STACK,$P ;LOAD THE STACK POINTER
8686
8687          ;CLEAR ATTENTION BITS FOR BOTH PORTS
8688
8689 060416 113760 001224 000010    MOVVB  PORTA,$MCS2($R0) ;SELECT PORT #A
8690 060424 005060 000012          CLR     $RMS1($R0)     ;SEIZE THE DRIVE
8691 060430 012760 000011 000000    MOV     #11,$RMS1($R0) ;ISSUE DRIVE CLEAR
8692 060436 012760 000013 000000    MOV     #13,$RMS1($R0) ;RELEASE THE DRIVE
8693 060444 113760 001226 000010    MOVVB  PORTB,$MCS2($R0) ;SELECT PORT #B
8694 060452 005060 000012          CLR     $RMS1($R0)     ;SEIZE THE DRIVE THROUGH PORT 'B'
8695 060456 012760 000011 000000    MOV     #11,$RMS1($R0) ;ISSUE DRIVE CLEAR
8696 060464 012760 000013 000000    MOV     #13,$RMS1($R0) ;RELEASE THE DRIVE
8697
8698          ;*****
8699
8700          ;SEIZE THE DRIVE THROUGH PORT B
8701
8702 060472 113760 001226 000010    MOVVB  PORTB,$MCS2($R0) ;SELECT PORT B
8703 060500 013737 001226 001242    MOV     PORTB,$SEIZPT  ;STORE SEIZING PORT'S ADDRESS
8704 060506 005060 000012          CLR     $RMS1($R0)     ;WRITE RMS1
8705 060512 013737 001224 001244    MOV     PORTA,$OPPR    ;'OPPOSITE' PORT ADDRESS
8706
8707          ;*****
8708          ;WAIT 500 MS
8709
8710
8711          ;*****
8712          ;START THE TIMER
8713
8714 060520 005037 001256          CLR     TIME          ;CLEAR THE ELAPSED TIME COUNTER
8715 060524 012737 000764 001260    MOV     #500.,WATCH   ;SET WATCH TO 500 MS
8716 060532 005737 001260 1$:  TST     WATCH        ;WATCH EQUAL TO ZERO
8717 060536 001375          BNE     1$           ;BR IF NOT
8718
8719          ;*****
8720          ;START THE TIMER
8721
8722 060540 005037 001256          CLR     TIME          ;CLEAR THE ELAPSED TIME COUNTER
8723 060544 012737 003720 001260    MOV     #2000.,WATCH  ;SET WATCH TO 2000 MS
8724
8725          ;*****
8726          ;RETRIGGER THE TIMEOUT ONE-SHOT
8727
8728 060552 005760 000012          TST     $RMS1($R0)    ;RETRIGGER THE ONE-SHOT
8729 060556 113760 001224 000010    MOVVB  PORTA,$MCS2($R0) ;SELECT PORT A
8730 060564 013737 001224 001240    MOV     PORTA,$PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8731 060572 005760 000012 2$:  TST     $RMS1($R0)    ;WAIT FOR TIMEOUT
8732 060576 001004          BNE     3$           ;BR IF TIMEOUT OCCURRED
8733 060600 005737 001260          TST     WATCH        ;WATCH EQUAL TO ZERO ?
  
```

```

8734 060604 001372          BNE      25          ;BR IF NOT
8735 060606 104036          ERROR   36          ;NO TIMEOUT WITHIN 2 SECONDS
8736 060610 013737 001256 001276 35:  MOV     TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
8737
8738 ;*****
8739
8740 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8741
8742 060616 005037 001254          CLR     RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
8743 060622 012737 000012 001122  MOV     #RMS1,$BODR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
8744 060630 060037 001122          ADD     R,$BODR     ;ADD THE I/O BASE ADDRESS
8745 060634 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8746 060642 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
8747 060650 016037 000012 001170  MOV     RMS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8748 060656 042737 024001 001170  BIC     #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
8749 060664 013737 001170 001164  MOV     STMP2,STMP0 ;COPY IT INTO 'STMP0'
8750 060672 042737 100100 001164  BIC     #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8751 060700 113760 0. 226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
8752 060706 016037 000012 001172  MOV     RMS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8753 060714 042737 024001 001172  BIC     #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
8754 060722 013737 001172 001166  MOV     STMP3,STMP1 ;COPY IT INTO 'STMP1'
8755 060730 042737 100100 001166  BIC     #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8756 060736 023737 001164 001166  CMP     STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8757 060744 001006          BNE     66$        ;BR IF NOT
8758 060746 005737 001164          TST     STMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8759 060752 001045          BNE     68$        ;BR IF NOT
8760 060754 104046          ERROR   46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8761 060756 000137 061142          JMP     70$        ;BYPASS THE REST OF THE CHECKS
8762 060762 013737 001170 001126 66$:  MOV     STMP2,$BODAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8763 060770 013737 001226 001240  MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8764 060776 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
8765 061004 005737 001164          TST     STMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
8766 061010 001414          BEQ     67$        ;BR IF ZERO
8767 061012 013737 001224 001240  MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8768 061020 013737 001172 001126  MOV     STMP3,$BODAT ;'BAD DATA' FOR ERROR TYPE OUT
8769 061026 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
8770 061034 005737 001166          TST     STMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
8771 061040 001012          BNE     68$        ;BR IF NOT
8772 061042 012737 177777 001254 67$:  MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8773 061050 012760 000011 000000  MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
8774 061056 012760 000013 000000  MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
8775 061064 104022          ERROR   22          ;TYPE ERROR MESSAGE 22
8776 061066 013737 001170 001126 68$:  MOV     STMP2,$BODAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8777 061074 013737 001224 001240  MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
8778 061102 023737 001124 001126  CMP     $GDDAT,$BODAT ;ALL BITS OK ?
8779 061110 001401          BEQ     69$        ;BR IF OK FROM PORT A.
8780 061112 104007          ERROR   7          ;REPORT ERROR
8781 061114 013737 001172 001126 69$:  MOV     STMP3,$BODAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
8782 061122 013737 001226 001240  MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
8783 061130 023737 001124 001126  CMP     $GDDAT,$BODAT ;SEE IF READ OK FROM PORT B.
8784 061136 001401          BEQ     70$        ;BR IF OK
8785 061140 104007          ERROR   7          ;REPORT ERROR
8786 061142 000240          70$:  NOP
8787
8788 ;*****
8789 ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

```

8790
8791 061144 023737 001276 001272      CMP      TIMES,TIMEBP      ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
8792 061152 003004                      BGT      45                ;BR IF GREATER
8793 061154 023737 001276 001274      CMP      TIMES,TIMEBM      ;MEASURED TIME LESS THAN -25% TOLERANCE
8794 061162 002001                      BGE      +4                ;BR IF NOT
8795 061164 104025                      45:     ERROR      25      ;REPORT THE ERROR
8796 061166 000004                      SCOPE                                ;LOOP ?
8797
8798
8799
8800

```

```

*****
*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 RMD51.
*
*  B.  SET PORT REQUEST BY WRITING 0'S INTO RMD51 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.
*
*****

```

```

8821 061170
8822 061170 005737 001300      TST      KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
8823 061174 001406                      BEQ      25                ;BR IF NOT
8824 061176 100002                      BPL      15                ;BR IF JUST ENTERED TEST
8825 061200 000137 002676      JMP      EXEC              ;RETURN & GET NEXT TEST NUMBER
8826 061204 012737 177777 001300 15:   MOV      #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
8827 061212 112737 000046 001102 25:   MOV     #46,$STSTNM        ;TEST NUMBER
8828 061220 012737 061242 001106      MOV      #TEST46,$LPADR    ;LOAD LOOP ON TEST ADDRESS
8829 061226 012737 061242 001110      MOV      #TEST46,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
8830 061234 012737 000002 001176      MOV      #2,$TIMES         ;DO 2. ITERATIONS
8831 061242 012706 001100      TEST46: MOV     #STACK,SP    ;LOAD THE STACK POINTER
8832
8833      ;CLEAR ATTENTION BITS FOR BOTH PORTS
8834
8835 061246 113760 001224 000010      MOV     PORTA,RMCS2(RO)    ;SELECT PORT #A
8836 061254 005060 000012          CLR     RMD51(RO)         ;SEIZE THE DRIVE
8837 061260 012760 000011 000000      MOV     #11,RMCS1(RO)     ;ISSUE DRIVE CLEAR
8838 061266 012760 000013 000000      MOV     #13,RMCS1(RO)     ;RELEASE THE DRIVE
8839 061274 113760 001226 000010      MOV     PORTB,RMCS2(RO)   ;SELECT PORT #B
8840 061302 005060 000012          CLR     RMD51(RO)         ;SEIZE THE DRIVE THROUGH PORT 'B'
8841 061306 012760 000011 000000      MOV     #11,RMCS1(RO)     ;ISSUE DRIVE CLEAR
8842 061314 012760 000013 000000      MOV     #13,RMCS1(RO)     ;RELEASE THE DRIVE
8843
8844
8845

```

H13

MC-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58 T46

MACY11 30(1046) 01-AUG-77 11:02 PAGE 163
PORT 'A' TIMEOUT/RELEASE TEST

SEQ 0165

```

8846                                     ;SEIZE THE DRIVE THROUGH PORT B
8847
8848 061322 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
8849 061330 013737 001226 001242      MOV   PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
8850 061336 005060 000012              CLR   RMDS1(RO)      ;WRITE RMDS1
8851 061342 013737 001224 001244      MOV   PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
8852 061350 113760 001224 000010      MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
8853 061356 013737 001224 001240      MOV   PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8854
8855 ;:*****
8856 ;SET REQUEST THROUGH PORT A
8857
8858 061364 005060 000012              CLR   RMDS1(RO)      ;SET REQUEST FOR PORT A
8859 061370 113760 001226 000010      MOVB  PORTB,RMCS2(RO) ;SELECT PORT B
8860 061376 013737 001226 001240      MOV   PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8861
8862 ;:*****
8863 ;RELEASE THE DRIVE THROUGH PORT B
8864
8865 061404 012760 000013 000000      MOV   #13,RMCS1(RO) ;RELEASE DRIVE THROUGH PORT B
8866
8867 ;:*****
8868 ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8869
8870 061412 013737 001264 001260      MOV   TIMEAP,WATCH  ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8871
8872 ;:*****
8873 ;VERIFY THAT THE DRIVE IS SEIZED BY PORT A
8874
8875 061420 005037 001250              CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8876 061424 016037 000012 001126      MOV   RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
8877 061432 012737 000012 001122      MOV   #RMDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8878 061440 060037 001122              ADD   RO,SBADR       ;ADD R11 BASE ADDRESS
8879 061444 005037 001124              CLR   $GDDAT         ;WHAT REGISTER SHOULD BE
8880 061450 023737 001124 001126      CMP   $GDDAT,SBDDAT  ;IS THE REGISTER OK ?
8881 061456 001403                      BEQ   66$            ;BR IF OK
8882 061460 104031                      ERROR 31             ;TYPE MESSAGE 31
8883 061462 005137 001250              COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8884 061466 000240                      NOP
8885 061470 005737 001250      66$: TST   CKERR          ;REGISTER OK ?
8886 061474 001402                      BEQ   .+6            ;BR IF OK
8887 061476 000137 062052                      JMP   1$             ;BYPASS REST OF TEST IF NOT
8888
8889
8890 ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
8891 061502 005737 001260      TST   WATCH          ;WATCH EQUAL ZERO ?
8892 061506 001375                      BNE   .-4            ;BR IF NOT
8893
8894 ;:*****
8895 ;CONFIRM THAT THE DRIVE HAS TIMED OUT
8896
8897
8898 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8899
8900 061510 005037 001254              CLR   RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8901 061514 012737 000012 001122      MOV   #RMDS1,SBADR   ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT

```

```

8902 061522 060037 001122 ADD RD, $BDADR ;ADD THE I/O BASE ADDRESS
8903 061526 012737 011700 001124 MOV #MOL!PGM!DPR!DRY:VV, $GDDAT ;COMPARISON CONSTANT
8904 061534 113760 001224 000010 MOV#B PORTA, RMCS2(R0) ;SELECT PORT A.
8905 061542 016037 000012 001170 MOV RMDS1(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8906 061550 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
8907 061556 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
8908 061564 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8909 061572 113760 001226 000010 MOV#B PORTB, RMCS2(R0) ;SELECT PORT B.
8910 061600 016037 000012 001172 MOV RMDS1(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8911 061606 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
8912 061614 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
8913 061622 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8914 061630 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8915 061636 001006 BNE 68$ ;BR IF NOT
8916 061640 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8917 061644 001045 BNE 70$ ;BR IF NOT
8918 061646 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8919 061650 000137 062050 JMP 72$ ;BYPASS THE REST OF THE CHECKS
8920 061654 013737 001170 001126 68$: MOV $TMP2, $BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8921 061662 013737 001226 001240 MOV#B PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8922 061670 113760 001226 000010 MOV#B PORTB, RMCS2(R0) ;SELECT PORT B.
8923 061676 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8924 061702 001414 BEQ 69$ ;BR IF ZERO
8925 061704 013737 001224 001240 MOV#B PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8926 061712 013737 001172 001126 MOV $TMP3, $BDAT ;'BAD DATA' FOR ERROR TYPE OUT
8927 061720 113760 001224 000010 MOV#B PORTA, RMCS2(R0) ;SELECT PORT A.
8928 061726 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8929 061732 001012 BNE 70$ ;BR IF NOT
8930 061734 012737 177777 001254 69$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
8931 061742 012760 000011 000000 MOV #11, RMCS1(R0) ;CLEAR THE DRIVE
8932 061750 012760 000013 000000 MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
8933 061756 104035 ERROR 35 ;TYPE ERROR MESSAGE 35
8934 061760 013737 001170 001126 70$: MOV $TMP2, $BDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8935 061766 013737 001224 001240 MOV#B PORTA, PTNBR ;CHANGE PORT NUMBER
8936 061774 042737 100000 001126 BIC #ATA, $BDAT ;DON'T CHECK THE ATTN BIT
8937 062002 023737 001124 001126 CMP $GDDAT, $BDAT ;ALL BITS OK ?
8938 062010 001401 BEQ 71$ ;BR IF OK FROM PORT A.
8939 062012 104007 ERROR 7 ;REPORT ERROR
8940 062014 013737 001172 001126 71$: MOV $TMP3, $BDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8941 062022 013737 001226 001240 MOV#B PORTB, PTNBR ;CHANGE PORT NUMBER
8942 062030 042737 100000 001126 BIC #ATA, $BDAT ;DON'T CHECK THE ATTN BIT
8943 062036 023737 001124 001126 CMP $GDDAT, $BDAT ;SEE IF READ OK FROM PORT B.
8944 062044 001401 BEQ 72$ ;BR IF OK
8945 062046 104007 ERROR 7 ;REPORT ERROR
8946 062050 000240 72$: NOP
8947 062052 000004 1$: SCOPE ;LOOP ?

```

```

8948
8949 ;*****
8950 ;*TEST 47 PORT 'B' TIMEOUT/RELEASE TEST
8951 ;*
8952 ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
8953 ;*SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
8954 ;*
8955 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
8956 ;*
8957 ;* B. SET PORT REQUEST BY WRITING 0'S INTO RMDS1 FROM PORT 'B'.

```

```

8958
8959
8960
8961
8962
8963
8964
8965
8966
8967 062054
8968 062054 005737 001300
8969 062060 001406
8970 062062 100002
8971 062064 000137 002676
8972 062070 012737 177777 001300
8973 062076 112737 000047 001102
8974 062104 012737 062126 001106
8975 062112 012737 062126 001110
8976 062120 012737 000002 001176
8977 062126 012706 001100
8978
8979
8980
8981 062132 113760 001224 000010
8982 062140 005060 000012
8983 062144 012760 000011 000000
8984 062152 012760 000013 000000
8985 062160 113760 001226 000010
8986 062166 005060 000012
8987 062172 012760 000011 000000
8988 062200 012760 000013 000000
8989
8990
8991
8992
8993
8994 062206 113760 001224 000010
8995 062214 013737 001224 001242
8996 062222 005060 000012
8997 062226 013737 001226 001244
8998 062234 113760 001226 000010
8999 062242 013737 001226 001240
9000
9001
9002
9003
9004 062250 005060 000012
9005 062254 113760 001224 000010
9006 062262 013737 001224 001240
9007
9008
9009
9010
9011 062270 012760 000013 000000
9012
9013

```

```

;*
;* 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.
;*
;*****
;ST47:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      25          ;BR IF NOT
BPL      15          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
15:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25:      MOVB     #47,$STNM   ;TEST NUMBER
        MOV      #TEST47,$LPADR ;LOAD LOOP ON TEST ADDRESS
        MOV      #TEST47,$LPER ;LOAD LOOP ON ERROR ADDRESS
        MOV      #2,$TIMES    ;DO 2. ITERATIONS
TEST47:  MOV      #STACK,SP  ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB     PORTA,RMCS2(RO) ;SELECT PORT #A
CLR      RMDS1(RO)      ;SEIZE THE DRIVE
MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
MOVB     PORTB,RMCS2(RO) ;SELECT PORT #B
CLR      RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE

;*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
MOV      PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
CLR      RMDS1(RO)      ;WRITE RMDS1
MOV      PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
MOV      PORTB,PTNBR    ;MOVE PCRT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;SET REQUEST THROUGH PORT B
CLR      RMDS1(RO)      ;SET REQUEST FOR PORT B
MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;RELEASE THE DRIVE THROUGH PORT A
MOV      #13,RMCS1(RO)  ;RELEASE DRIVE THROUGH PORT A

;*****

```

```

9014 ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
9015
9016 062276 013737 001272 001260 MOV TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
9017
9018 ;*****
9019 ;VERIFY THAT THE DRIVE IS SEIZED BY PORT B
9020
9021 062304 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9022 062310 016037 000012 001126 MOV RMD51(RO),SBDDAT ;GET CONTENTS OF RMD51
9023 062316 012737 000012 001122 MOV #RMD51,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9024 062324 060037 001122 ADD RO,SBADR ;ADD RHI1 BASE ADDRESS
9025 062330 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
9026 062334 023737 001124 001126 CMP $GDDAT,SBDDAT ;IS THE REGISTER OK ?
9027 062342 001403 BEQ 66$ ;BR IF OK
9028 062344 104031 ERROR 31 ;TYPE MESSAGE 31
9029 062346 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9030 062352 000240 66$: NOP
9031 062354 005737 001250 TST CKERR ;REGISTER OK ?
9032 062360 001402 BEQ .+6 ;BR IF OK
9033 062362 000137 062736 JMP 1$ ;BYPASS REST OF TEST IF NOT
9034
9035
9036 ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
9037 062366 005737 001260 TST WATCH ;WATCH EQUAL ZERO ?
9038 062372 001375 BNE .-4 ;BR IF NOT
9039
9040 ;*****
9041 ;CONFIRM THAT THE DRIVE HAS TIMED OUT
9042
9043
9044 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
9045
9046 062374 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
9047 062400 012737 000012 001122 MOV #RMD51,SBADR ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
9048 062406 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
9049 062412 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
9050 062420 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
9051 062426 016037 000012 001170 MOV RMD51(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
9052 062434 042737 024001 001170 BIC #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
9053 062442 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
9054 062450 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9055 062456 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9056 062464 016037 000012 001172 MOV RMD51(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
9057 062472 042737 024001 001172 BIC #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
9058 062500 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
9059 062506 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9060 062514 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
9061 062522 001006 BNE 68$ ;BR IF NOT
9062 062524 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
9063 062530 001045 BNE 70$ ;BR IF NOT
9064 062532 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
9065 062534 000137 062734 JMP 72$ ;BYPASS THE REST OF THE CHECKS
9066 062540 013737 001170 001126 68$: MOV STMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9067 062546 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9068 062554 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9069 062562 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.

```



```

9070 062566 001414 BEQ 69$ ;BR IF ZERO
9071 062570 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9072 062576 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
9073 062604 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
9074 062612 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
9075 062616 001012 BNE 70$ ;BR IF NOT
9076 062620 012737 177777 001254 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
9077 062626 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
9078 062634 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
9079 062642 104035 ERROR 35 ;TYPE ERROR MESSAGE 35
9080 062644 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
9081 062652 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
9082 062660 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
9083 062666 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
9084 062674 001401 BEQ 71$ ;BR IF OK FROM PORT A.
9085 062676 104007 ERROR 7 ;REPORT ERROR
9086 062700 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
9087 062706 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
9088 062714 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
9089 062722 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
9090 062730 001401 BEQ 72$ ;BR IF OK
9091 062732 104007 ERROR 7 ;REPORT ERROR
9092 062734 000240 72$: NOP
9093 062736 000004 1$: SCOPE ;LOOP ?

```

```

9094
9095
9096 *****
9097 *TEST 50 PORT 'A' SEIZE ACCESS TEST
9098 *
9099 *VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
9100 *
9101 * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
9102 *
9103 * B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.
9104 *
9105 * C. READ RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT
9106 * 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
9107 *
9108 * D. CLEAR RMER1, RMER2 THROUGH PORT 'A'.
9109 *
9110 * E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT
9111 * PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
9112 *
9113 * F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS
9114 * SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
9115 * SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
9116 *
9117 * G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
9118 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
9119 *
9120 *****
9121 †T50: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
9122 062740 005737 001300 BEQ 2$ ;BR IF NOT
9123 062744 001406 BPL 1$ ;BR IF JUST ENTERED TEST
9124 062746 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
9125 062750 000137 002676

```

M13

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T50

MACY11 30(1046) 01-AUG-77 11:02 PAGE 168
 PORT 'A' SEIZE ACCESS TEST

SEQ 0170

```

9126 062754 012737 177777 001300 15:  MOV    #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
9127 062762 112737 000050 001102 25:  MOVVB  #50,$TSTNM      ;TEST NUMBER
9128 062770 012737 063012 001106      MOV    #TEST50,$LPADR  ;LOAD LOOP ON TEST ADDRESS
9129 062776 012737 063012 001110      MOV    #TEST50,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
9130 063004 012737 000031 001176      MOV    #25,$TIMES     ;DO 25. ITERATIONS
9131 063012 012706 001100      TEST50: MOV    #STACK,SP  ;LOAD THE STACK POINTER
9132
9133      ;CLEAR ATTENTION BITS FOR BOTH PORTS
9134
9135 063016 113760 001224 000010      MOVVB  PORTA, RMCS2(RO) ;SELECT PORT #A
9136 063024 005060 000012      CLR    RMDS1(RO)       ;SEIZE THE DRIVE
9137 063030 012760 000011 000000      MOV    #11, RMCS1(RO)  ;ISSUE DRIVE CLEAR
9138 063036 012760 000013 000000      MOV    #13, RMCS1(RO)  ;RELEASE THE DRIVE
9139 063044 113760 001226 000010      MOVVB  PORTB, RMCS2(RO) ;SELECT PORT #B
9140 063052 005060 000012      CLR    RMDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
9141 063056 012760 000011 000000      MOV    #11, RMCS1(RO)  ;ISSUE DRIVE CLEAR
9142 063064 012760 000013 000000      MOV    #13, RMCS1(RO)  ;RELEASE THE DRIVE
9143
9144      ;SEIZE THE DRIVE THROUGH PORT A
9145
9146 063072 113760 001224 000010      MOVVB  PORTA, RMCS2(RO) ;SELECT PORT A
9147 063100 013737 001224 001242      MOV    PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
9148 063106 005060 000012      CLR    RMDS1(RO)       ;WRITE RMDS1
9149 063112 013737 001226 001244      MOV    PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
9150 063120 012760 177777 000014      MOV    #-1, RMER1(RO)  ;LOAD 1'S INTO RMER1 THROUGH PORT A
9151 063126 012760 177777 000042      MOV    #-1, RMER2(RO)  ;LOAD 1'S INTO RMER2 THROUGH PORT A
9152 063134 113760 001226 000010      MOVVB  PORTB, RMCS2(RO) ;SELECT PORT B
9153 063142 013737 001226 001240      MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9154 063150 004737 064010      JSR    PC, TST50B     ;CHECK THE REGISTERS THROUGH PORT B
9155 063154 113760 001224 000010      MOVVB  PORTA, RMCS2(RO) ;SELECT PORT A
9156 063162 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9157 063170 005060 000042      CLR    RMER2(RO)       ;CLEAR RMER2 ON PORT A
9158 063174 005060 000014      CLR    RMER1(RO)       ;CLEAR RMER1 ON PORT A
9159 063200 013760 001236 000016      MOV    ASR1, RMAS(RO)  ;CLEAR THE ATTENTION BIT FOR PORT A
9160 063206 113760 001226 000010      MOVVB  PORTB, RMCS2(RO) ;SELECT PORT B
9161 063214 013737 001226 001240      MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9162 063222 012760 177777 000014      MOV    #-1, RMER1(RO)  ;LOAD 1'S INTO RMER1 THROUGH PORT B
9163 063230 012760 177777 000042      MOV    #-1, RMER2(RO)  ;LOAD 1'S INTO RMER2 THROUGH PORT B
9164 063236 113760 001224 000010      MOVVB  PORTA, RMCS2(RO) ;SELECT PORT A
9165 063244 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9166 063252 004737 064010      JSR    PC, TST50B     ;CHECK THE REGISTERS THROUGH PORT A
9167
9168      ;RELEASE THE DRIVE FROM PORT A
9169
9170 063256 113760 001224 000010      MOVVB  PORTA, RMCS2(RO) ;SELECT PORT A
9171 063264 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9172 063272 012760 000013 000000      MOV    #13, RMCS1(RO)  ;ISSUE RELEASE THROUGH PORT A
9173
9174      ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
9175
9176 063300 005037 001254      CLR    RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
9177 063304 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV,$GDOAT ;COMPARISON CONSTANT
9178 063312 012737 000012 001122      MOV    #RMDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
9179 063320 060037 001122      ADD    RO, $BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
9180 063324 113760 001226 000010      MOVVB  PORTB, RMCS2(RO) ;SELECT PORT B
9181 063332 013737 001226 001240      MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

9182	063340	016037	000012	001164		MOV	RMS1(RO), \$TMP0	; READ STATUS REGISTER FROM PORT B
9183	063346	113760	001224	000010		MOVB	PORTA, RMS2(RO)	; SELECT PORT A
9184	063354	013737	001224	001240		MOV	PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9185	063362	016037	000012	001126		MOV	RMS1(RO), \$BDDAT	; DRIVE STATUS FROM PORT A
9186	063370	001404				BEQ	66\$; BR IF STATUS FROM PORT A ZERO
9187	063372	005737	001164			TST	\$TMP0	; IS STATUS FROM PORT B ZERO ?
9188	063376	001401				BEQ	66\$; BR IF ZERO
9189	063400	104031				ERROR	31	; REPORT DRIVE IN NEUTRAL
9190	063402	013737	001164	001126	66\$:	MOV	\$TMP0, \$BDDAT	; CHECK STATUS FROM PORT B
9191	063410	013737	001226	001240		MOV	PORTB, PTNBR	; CHANGE PORT ADDRESS FOR TYPEOUT
9192	063416	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; COMPARE WITH CONSTANT
9193	063424	001401				BEQ	67\$; BR IF OK
9194	063426	104027				ERROR	27	; REPORT REGISTER ERROR
9195	063430	000240			67\$:	NOP		
9196								
9197								; RELEASE THE DRIVE FROM PORT B
9198								
9199	063432	113760	001226	000010		MOVB	PORTB, RMS2(RO)	; SELECT PORT B
9200	063440	013737	001226	001240		MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9201	063446	012760	000013	000000		MOV	#13, RMS1(RO)	; ISSUE RELEASE THROUGH PORT B
9202								
9203								; VERIFY THAT THE DRIVE IS IN NEUTRAL
9204								
9205	063454	005037	001254			CLR	RELERR	; CLEAR THE 'RELEASE ERROR' INDICATOR
9206	063460	012737	000012	001122		MOV	#RMS1, \$BDDADR	; FORM THE ADDRESS OF RMS1 FOR TYPEOUT
9207	063466	060037	001122			ADD	RO, \$BDDADR	; ADD THE I/O BASE ADDRESS
9208	063472	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	; COMPARISON CONSTANT
9209	063500	113760	001224	000010		MOVB	PORTA, RMS2(RO)	; SELECT PORT A.
9210	063506	016037	000012	001170		MOV	RMS1(RO), \$TMP2	; GET THE DRIVE STATUS REGISTER FROM PORT A.
9211	063514	042737	024001	001170		BIC	#PIP!WRL!OM, \$TMP2	; CLEAR DONT CARES
9212	063522	013737	001170	001164		MOV	\$TMP2, \$TMP0	; COPY IT INTO 'TMP0'
9213	063530	042737	100100	001164		BIC	#ATA!VV, \$TMP0	; CLEAR PORT DEPENDENT BITS FROM THE COPY
9214	063536	113760	001226	000010		MOVB	PORTB, RMS2(RO)	; SELECT PORT B.
9215	063544	016037	000012	001172		MOV	RMS1(RO), \$TMP3	; GET THE DRIVE STATUS REGISTER FROM PORT B.
9216	063552	042737	024001	001172		BIC	#PIP!WRL!OM, \$TMP3	; CLEAR DONT CARES
9217	063560	013737	001172	001166		MOV	\$TMP3, \$TMP1	; COPY IT INTO 'TMP1'
9218	063566	042737	100100	001166		BIC	#ATA!VV, \$TMP1	; CLEAR PORT DEPENDENT BITS FROM THE COPY
9219	063574	023737	001164	001166		CMP	\$TMP0, \$TMP1	; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
9220	063602	001106				BNE	68\$; BR IF NOT
9221	063604	005737	001164			TST	\$TMP0	; REGISTERS ARE THE SAME: ARE THEY ZERO ?
9222	063610	001045				BNE	70\$; BR IF NOT
9223	063612	104045				ERROR	46	; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
9224	063614	000137	064000			JMP	72\$; BYPASS THE REST OF THE CHECKS
9225	063620	013737	001170	001126	68\$:	MOV	\$TMP2, \$BDDAT	; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9226	063626	013737	001226	001240		MOV	PORTB, PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9227	063634	113760	001226	000010		MOVB	PORTB, RMS2(RO)	; SELECT PORT B.
9228	063642	005737	001164			TST	\$TMP0	; SEE IF STATUS EQ 0 FROM PORT A.
9229	063646	001414				BEQ	69\$; BR IF ZERO
9230	063650	013737	001224	001240		MOV	PORTA, PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9231	063656	013737	001172	001126		MOV	\$TMP3, \$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
9232	063664	113760	001224	000010		MOVB	PORTA, RMS2(RO)	; SELECT PORT A.
9233	063672	005737	001166			TST	\$TMP1	; SEE IF STATUS EQ ZERO FROM PORT B.
9234	063676	001012				BNE	70\$; BR IF NOT
9235	063700	012737	177777	001254	69\$:	MOV	#-1, RELERR	; SET 'RELEASE ERROR' INDICATOR
9236	063706	012760	000011	000000		MOV	#11, RMS1(RO)	; CLEAR THE DRIVE
9237	063714	012760	000013	000000		MOV	#13, RMS1(RO)	; RELEASE THE DRIVE

```

9238 063722 104026          ERROR 26          ;TYPE ERROR MESSAGE 26
9239 063724 013737 001170 001126 70S:  MOV  $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD51 READ
9240 063732 012737 001224 001240      MOV  PORTA,PTNBR ;CHANGE PORT NUMBER
9241 063740 023737 001124 001126      CMP  $GDDAT,$BDDAT ;ALL BITS OK ?
9242 063746 001401          BEQ  71S          ;BR IF OK FROM PORT A.
9243 063750 104007          ERROR 7           ;REPORT ERROR
9244 063752 013737 001172 001126 71S:  MOV  $TMP3,$BDDAT ;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
9245 063760 013737 001226 001240      MOV  PORTB,PTNBR ;CHANGE PORT NUMBER
9246 063766 023737 001124 001126      CMP  $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
9247 063774 001401          BEQ  72S          ;BR IF OK
9248 063776 104007          ERROR 7           ;REPORT ERROR
9249 064000 000240          72S:  NOP
9250 064002 000004          SCOPE
9251 064004 000137 064232      JMP  TST51       ;LOOP ?
9252                                     ;GO TO THE NEXT TEST
9253                                     ;CHECK THE REGISTERS ON THE SELECTED PORT
9254
9255                                     TST50B:
9256 064010 005037 001250          CLR  CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
9257 064014 016037 000014 001126      MOV  RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
9258 064022 012737 000014 001122      MOV  #RMER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9259 064030 060037 001122          ADD  RO,$B0ADR   ;ADD RH11 BASE ADDRESS
9260 064034 005037 001124          CLR  $GDDAT      ;WHAT REGISTER SHOULD BE
9261 064040 023737 001124 001126      CMP  $GDDAT,$BDDAT ;IS THE REGISTER OK ?
9262 064046 001403          BEQ  64S          ;BR IF OK
9263 064050 104006          ERROR 6         ;TYPE MESSAGE 6
9264 064052 005137 001250          COM  CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
9265 064056 016037 000000 001126 64S:  MOV  RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
9266 064064 012737 000000 001122      MOV  #RMCS1,$B0ADR ;FORM ADDRESS OF REGISTER
9267 064072 060037 001122          ADD  RO,$B0ADR   ;ADDRESS BASE
9268 064076 032737 020000 001126      BIT  #MCPE,$BDDAT ;IS 'MCPE' SET ?
9269 064104 001404          BEQ  65S          ;BR IF NOT
9270 064106 104011          ERROR 11        ;REPORT THE ERROR
9271 064110 012760 040000 000000      MOV  #TRE,RMCS1(RO) ;CLEAR 'MCPE'
9272 064116 000240          65S:  NOP
9273 064120 005037 001250          CLR  CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
9274 064124 016037 000042 001126      MOV  RMER2(RO),$BDDAT ;GET CONTENTS OF RMER2
9275 064132 012737 000042 001122      MOV  #RMER2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9276 064140 060037 001122          ADD  RO,$B0ADR   ;ADD RH11 BASE ADDRESS
9277 064144 005037 001124          CLR  $GDDAT      ;WHAT REGISTER SHOULD BE
9278 064150 023737 001124 001126      CMP  $GDDAT,$BDDAT ;IS THE REGISTER OK ?
9279 064156 001403          BEQ  66S          ;BR IF OK
9280 064160 104006          ERROR 6         ;TYPE MESSAGE 6
9281 064162 005137 001250          COM  CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
9282 064166 016037 000000 001126 66S:  MOV  RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
9283 064174 012737 000000 001122      MOV  #RMCS1,$B0ADR ;FORM ADDRESS OF REGISTER
9284 064202 060037 001122          ADD  RO,$B0ADR   ;ADDRESS BASE
9285 064206 032737 020000 001126      BIT  #MCPE,$BDDAT ;IS 'MCPE' SET ?
9286 064214 001404          BEQ  67S          ;BR IF NOT
9287 064216 104011          ERROR 11        ;REPORT THE ERROR
9288 064220 012760 040000 000000      MOV  #TRE,RMCS1(RO) ;CLEAR 'MCPE'
9289 064226 000240          67S:  NOP
9290 064230 000207          RTS  PC          ;RETURN
9291
9292                                     ;*****
9293                                     ;*TEST 51      PORT 'B' SEIZE ACCESS TEST

```

```

9294
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317 064232
9318 064232 005737 001300
9319 064236 001406
9320 064240 100002
9321 064242 000137 002676
9322 064246 012737 177777 001300
9323 064254 112737 000051 001102
9324 064262 012737 064304 001106
9325 064270 012737 064304 001110
9326 064276 012737 000031 001176
9327 064304 012706 001100
9328
9329
9330
9331 064310 113760 001224 000010
9332 064316 005060 000012
9333 064322 012760 000011 000000
9334 064330 012760 000013 000000
9335 064336 113760 001226 000010
9336 064344 005060 000012
9337 064350 012760 000011 000000
9338 064356 012760 000013 000000
9339
9340
9341
9342 064364 113760 001226 000010
9343 064372 013737 001226 001242
9344 064400 005060 000012
9345 064404 013737 001224 001244
9346 064412 012760 177777 000014
9347 064420 012760 177777 000042
9348 064426 113760 001224 000010
9349 064434 013737 001224 001240

```

```

;*
;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
;*
;* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
;*
;* 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.
;*
*****
TST51:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #51, STSTNM ;TEST NUMBER
MOV #TEST51, $LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST51, $LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25, $TIMES ;DO 25. ITERATIONS
TEST51: MOV #STACK, SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA, RMCS2(RO) ;SELECT PORT #A
CLR RMDS1(RO) ;SEIZE THE DRIVE
MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB, RMCS2(RO) ;SELECT PORT #B
CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RMCS1(RO) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDS1(RO) ;WRITE RMDS1
MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
MOV #-1, RMER1(RO) ;LOAD 1'S INTO RMER1 THROUGH PORT B
MOV #-1, RMER2(RO) ;LOAD 1'S INTO RMER2 THROUGH PORT B
MOVB PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

9350 064442 004737 065302 JSR PC,T51B ;CHECK THE REGISTERS THROUGH PORT A
9351 064446 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
9352 064454 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9353 064462 005060 000042 CLR RMER2(RO) ;CLEAR RMER2 ON PORT B
9354 064466 005060 000014 CLR RMER1(RO) ;CLEAR RMER1 ON PORT B
9355 064472 013760 001236 000016 MOV ASR1,RMAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT B
9356 064500 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
9357 064506 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9358 064514 012760 177777 000014 MOV #-1,RMER1(RO) ;LOAD 1'S INTO RMER1 THROUGH PORT A
9359 064522 012760 177777 000042 MOV #-1,RMER2(RO) ;LOAD 1'S INTO RMER2 THROUGH PORT A
9360 064530 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
9361 064536 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9362 064544 004737 065302 JSR PC,T51B ;CHECK THE REGISTERS THROUGH PORT B
9363
9364 ;RELEASE THE DRIVE FROM PORT B
9365
9366 064550 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
9367 064556 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9368 064564 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
9369
9370 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
9371
9372 064572 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
9373 064576 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
9374 064604 012737 000012 001122 MOV #RMDS1,$B0ADR ;REGISTER ADDRESS INCREMENT
9375 064612 060037 001122 ADD RO,$B0ADR ;REGISTER BASE ADDRESS FOR TYPEOUT
9376 064616 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
9377 064624 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9378 064632 016037 000012 001164 MOV RMDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
9379 064640 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
9380 064646 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9381 064654 016037 000012 001126 MOV RMDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
9382 064662 001404 BEQ 66$ ;BR IF STATUS FROM PORT B ZERO
9383 064664 005737 001164 TST $TMP0 ;IS STATUS FROM PORT A ZERO ?
9384 064670 001401 BEQ 66$ ;BR IF ZERO
9385 064672 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
9386 064674 013737 001164 001126 66$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
9387 064702 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
9388 064710 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
9389 064716 001401 BEQ 67$ ;BR IF OK
9390 064720 104027 ERROR 27 ;REPORT REGISTER ERROR
9391 064722 000240 67$: NOP
9392
9393 ;RELEASE THE DRIVE FROM PORT A
9394
9395 064724 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
9396 064732 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9397 064740 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
9398
9399 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
9400
9401 064746 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
9402 064752 012737 000012 001122 MOV #RMDS1,$B0ADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
9403 064760 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
9404 064764 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
9405 064772 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
    
```

E14

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58 T51

MACY11 30(1046) 01-AUG-77 11:02 PAGE 173
 PORT 'B' SEIZE ACCESS TEST

SEQ 0175

```

9406 065000 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
9407 065006 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
9408 065014 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
9409 065022 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9410 065030 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9411 065036 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
9412 065044 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
9413 065052 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
9414 065060 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9415 065066 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
9416 065074 001006 BNE 68$ ;BR IF NOT
9417 065076 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
9418 065102 001045 BNE 70$ ;BR IF NOT
9419 065104 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
9420 065106 000137 065272 JMP 72$ ;BYPASS THE REST OF THE CHECKS
9421 065112 013737 001170 001126 68$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9422 065120 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9423 065126 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9424 065134 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
9425 065140 001414 BEQ 69$ ;BR IF ZERO
9426 065142 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9427 065150 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
9428 065156 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
9429 065164 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
9430 065170 001012 BNE 70$ ;BR IF NOT
9431 065172 012737 177777 001254 69$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
9432 065200 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
9433 065206 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
9434 065214 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
9435 065216 013737 001170 001126 70$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
9436 065224 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
9437 065232 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
9438 065240 001401 BEQ 71$ ;BR IF OK FROM PORT A.
9439 065242 104007 ERROR 7 ;REPORT ERROR
9440 065244 013737 001172 001126 71$: MOV $TMP3, $BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
9441 065252 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
9442 065260 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
9443 065266 001401 BEQ 72$ ;BR IF OK
9444 065270 104007 ERROR 7 ;REPORT ERROR
9445 065272 000240 72$: NOP ;LOOP ?
9446 065274 000004 SCOPE ;GO TO THE NEXT TEST
9447 065276 000137 065524 JMP TST52
9448
9449 ;CHECK THE REGISTERS ON THE SELECTED PORT
9450
9451 TST51B:
9452 065302 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9453 065306 016037 000014 001126 MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
9454 065314 012737 000014 001122 MOV #RMER1, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9455 065322 060037 001122 ADD RO, $BDDADR ;ADD RH11 BASE ADDRESS
9456 065326 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
9457 065332 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
9458 065340 001403 BEQ 64$ ;BR IF OK
9459 065342 104006 ERROR 6 ;TYPE MESSAGE 6
9460 065344 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9461 065350 016037 000000 001126 64$: MOV RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
    
```

```

9462 065356 012737 000000 001122      MOV      #RMCS1,$BDADR ;FORM ADDRESS OF REGISTER
9463 065364 060037 001122      ADD      R0,$BDADR ;ADDRESS BASE
9464 065370 032737 020000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET ?
9465 065376 001404      BEQ      65$ ;BR IF NOT
9466 065400 104011      ERROR   11 ;REPORT THE ERROR
9467 065402 012760 040000 000000      MOV      #TRE,RMCS1(R0) ;CLEAR 'MCPE'
9468 065410 000240      NOP
9469 065412 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9470 065416 016037 000042 001126      MOV      RMER2(R0),$BDDAT ;GET CONTENTS OF RMER2
9471 065424 012737 000042 001122      MOV      #RMER2,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9472 065432 060037 001122      ADD      R0,$BDADR ;ADD RH11 BASE ADDRESS
9473 065436 005037 001124      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
9474 065442 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS IT REGISTER OK ?
9475 065450 001403      BEQ      66$ ;BR IF OK
9476 065452 104006      ERROR   6 ;TYPE MESSAGE 6
9477 065454 005137 001250      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9478 065460 016037 000000 001126      MOV      RMCS1(R0),$BDDAT ;GET THE CONTENTS OF RMCS1
9479 065466 012737 000000 001122      MOV      #RMCS1,$BDADR ;FORM ADDRESS OF REGISTER
9480 065474 060037 001122      ADD      R0,$BDADR ;ADDRESS BASE
9481 065500 032737 020000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET ?
9482 065506 001404      BEQ      67$ ;BR IF NOT
9483 065510 104011      ERROR   11 ;REPORT THE ERROR
9484 065512 012760 040000 000000      MOV      #TRE,RMCS1(R0) ;CLEAR 'MCPE'
9485 065520 000240      NOP
9486 065522 000207      RTS      PC ;RETURN

```

```

9487
9488      ;*****
9489 065524 000004      †T52: SCOPE
9490
9491      .SBTTL END OF PASS ROUTINE
9492
9493      ;*****
9494      ;INCREMENT THE PASS NUMBER ($PASS)
9495      ;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
9496      ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYY"
9497      ;*WHERE XXXXX AND YYYY ARE DECIMAL NUMBERS
9498      ;*IF THERES A MONITOR GO TO IT
9499      ;*IF THERE ISN'T JUMP TO TST1AA
9500

```

```

9501      $EOP:
9502 065526 005737 001300      TST      KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
9503 065532 001402      BEQ      .+6 ;BR IF NOT
9504 065534 000137 002676      JMP      EXEC ;RETURN TO KEYBOARD CONTROL
9505 065540 005037 001102      CLR      $.STNM ;ZERO THE TEST NUMBER
9506 065544 005037 001176      CLR      $TIMES ;ZERO THE NUMBER OF ITERATIONS
9507 065550 005237 001100      INC      $PASS ;INCREMENT THE PASS NUMBER
9508 065554 042737 100000 001100      BIC      #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
9509 065562 005327      DEC      (PC)+ ;LOOP?
9510 065564 000001      $EOPCT: .WORD 1
9511 065566 003063      BGT      $DOAGN ;;YES
9512 065570 012737      MOV      (PC)+,2(PC)+ ;;RESTORE COUNTER
9513 065572 000001      $ENDCT: .WORD 1
9514 065574 065564      $EOPCT
9515 065576 104401 065604      TYPE    65$ ;;TYPE ASCIZ STRING
9516 065602 000407      BR      64$ ;;GET OVER THE ASCIZ
9517      ;;65$: .ASCIZ <12><15>/END PASS #/

```



```

9518 065622          645:      MOV      $PASS,-(SP)      ;;SAVE $PASS FOR TYPEOUT
9519 065622 013746 001100          ;;TYPE PASS NUMBER
9520          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
9521 065626 104405          TYPE      ,67$          ;;TYPE ASCIZ STRING
9522 065630 104401 065636          BR       ,66$          ;;GET OVER THE ASCIZ
9523 065634 000421          ;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
9524          66$:
9525 065700          MOV      $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
9526 065700 013746 001112          ;;TOTAL NUMBER OF ERRORS
9527          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
9528 065704 104405          TYPE      ,SCLRF        ;;TYPE CARRIAGE RETURN, LINE FEED
9529 065706 104401 001207          CLR      $ERTTL        ;;CLEAR ERROR TOTAL
9530 065712 005037 001112          $GET42: MOV     ,R0        ;;GET MONITOR ADDRESS
9531 065716 013700 000042          BEQ     $DOAGN        ;;BRANCH IF NO MONITOR
9532 065722 001405          RESET          ;;CLEAR THE WORLD
9533 065724 000005          $ENDAD: JSR    PC,(R0)  ;;GO TO MONITOR
9534 065726 004710          NOP          ;;SAVE ROOM
9535 065730 000240          NOP          ;;FOR
9536 065732 000240          NOP          ;;ACT11
9537 065734 000240          $DOAGN:
9538 065736          JMP     ,(PC)+        ;;RETURN
9539 065736 000137          $RTNAD: .WORD   TST1AA
9540 065740 003160          $ENULL: .BYTE  -1,-1,0  ;;NULL CHARACTER STRING
9541 065742          377      377      000
9542          065746
9543
9544          ;;*****
9545          .SBTTL  *** SUBROUTINES ***
9546
9547          ;;*****
9548
9549          ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
9550          ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
9551
9552          CKCLK: MOV     #CKCLK1,,$ERRVEC  ;;SET UP VECTOR FOR CLOCK CHECK
9553          CLR     ,,$ERRVEC+2              ;;NEW PSW
9554 065746 012737 066016 000004          TST     ,,$KCSR          ;;CHECK FOR KW11-P
9555 065754 005037 000006          MOV     $LPVEC,R1        ;;KW11-P VECTOR ADDRESS
9556 065760 005777 113226          MOV     #CLOCK,(R1)+    ;;SET UP KW11-P VECTOR
9557 065764 013701 001216          MOV     #300,(R1)      ;;PSW - PRI 6
9558 065770 012721 066100          MOV     #-1,,$SLKCSB    ;;LOAD COUNTER BUFFER WITH 1'S
9559 065774 012711 000300          MOV     #135,,$SLKCSR   ;;SET CLOCK - CNT UP, 16MS, CONT INT
9560 066000 012777 177777 113206          BR      CKCLK3
9561 066006 012777 000135 113176          CKCLK1: ADD     #4,SP    ;;RESTORE THE STACK POINTER
9562 066014 000425          MOV     #CKCLK2,,$ERRVEC ;;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
9563 066016 062706 000004          TST     ,,$SLKS        ;;LOOK FOR KW11-L
9564 066022 012737 066060 000004          MOV     $LLVEC,R1      ;;KW11-L VECTOR ADDRESS
9565 066030 005777 113164          MOV     #CLOCK,(R1)+   ;;SET UP KW11-L VECTOR
9566 066034 013701 001222          MOV     #300,(R1)      ;;PSW - PRI 6
9567 066040 012721 066100          MOV     #100,,$SLKS    ;;SET KW11-L INTERRUPT
9568 066044 012711 000300          BR      CKCLK3
9569 066050 012777 000100 113142          CKCLK2: ADD     #4,SP    ;;RESTORE THE STACK POINTER
9570 066056 000404          ADD     #2,(SP)        ;;INCREMENT RETURN, NO CLOCK
9571 066060 062706 000004          CKCLK3: MOV     #6,,$ERRVEC ;;RESTORE THE ERROR VECTOR
9572 066064 062716 000002
9573 066070 012737 000006 000004

```

*** SUBROUTINES ***

```

9574 066076 000207          RTS      PC
9575
9576          ;ROUTINE TO COUNT CLOCK TICKS
9577
9578 066100 062737 000021 001256  CLOCK:  ADD    #17.,TIME      ;ADD 17 MS TO ELAPSED TIME COUNTER
9579 066106 103003          BCC    1$          ;BRANCH IF NO OVERFLOW
9580 066110 012737 177777 001256          MOV    #-1,TIME    ;OVERFLOW - RESTORE MAXIMUM COUNT
9581 066116 005737 001260          1$:   TST    WATCH    ;IS WATCH ALREADY ZERO ?
9582 066122 001406          BEQ    2$          ;BR IF IT IS
9583 066124 162737 000021 001260          SUB    #17.,WATCH  ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
9584 066132 100002          BPL    2$          ;BR IF NOT MINUS
9585 066134 005037 001260          CLR    WATCH      ;CLEAR WATCH DOG COUNTER
9586 066140 000002          2$:   RTI          ;RETURN
9587
9588          ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
9589
9590 066142 162706 000004          TOLER: SUB    #4,SP      ;SETUP STACK
9591 066146 016616 000004          MOV    4(SP), (SP)    ;SAVE STACK
9592 066152 013546          MOV    @ (R5)+, -(SP) ;GET TIME VALUE
9593 066154 011666 000004          MOV    (SP), 4(SP)   ;MOVE TIME VALUE
9594 066160 011666 000006          MOV    (SP), 6(SP)   ;MOVE VALUE AGAIN
9595 066164 006216          ASR    (SP)          ;DIVIDE BY 2
9596 066166 006216          ASR    (SP)          ;DIVIDE BY 2 GAIN (FOR A TOTAL OF 4)
9597 066170 061666 000004          ADD    (SP), 4(SP)   ;CALCULATE UPPER LIMIT FOR TIMEOUT
9598 066174 162666 000004          SUB    (SP)+, 4(SP)  ;CALCULATE LOWER LIMIT FOR TIMEOUT
9599 066200 000205          RTS      R5          ;RETURN WITH TOLERANCES ON THE STACK
9600
9601          ;*****
9602
9603          .SBTTL 'SYSMAC' UTILITY ROUTINES
9604
9605          ;*****
9606
9607          .SBTTL SCOPE HANDLER ROUTINE
9608
9609          ;*****
9610          ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9611          ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<?:0>)
9612          ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9613          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9614          ;*SW14=1      LOOP ON TEST
9615          ;*SW11=1      INHIBIT ITERATIONS
9616          ;*SW09=1      LOOP ON ERROR
9617          ;*CALL
9618          ;*      SCOPE          ;;SCOPE=IOT
9619
9620          $SCOPE:
9621 066202 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
9622 066204 032777 040000 112726  1$:   BIT    #BIT14, @SWR ;;LOOP ON PRESENT TEST?
9623 066212 001101          BNE    $OVER      ;;YES IF SW14=1
9624          ;*****START OF CODE FOR THE XOR TESTER*****
9625 066214 000416          $XTSTR: BR    6$   ;;IF RUNNING ON THE "XOR" TESTER CHANGE
9626          ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
9627 066216 013746 000004          MOV    @#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
9628 066222 012737 066242 000004          MOV    #5$, @#ERRVEC ;;SET FOR TIMEOUT
9629 066230 005737 177060          TST    @#177060    ;;TIME OUT ON XOR?

```

```

9630 066234 012637 000004      MOV      (SP)+, @ERRVEC      ;; RESTORE THE ERROR VECTOR
9631 066240 000453      BR       $SVLAD             ;; GO TO THE NEXT TEST
9632 066242 022626      5$:     CMP      (SP)+, (SP)+      ;; CLEAR THE STACK AFTER A TIME OUT
9633 066244 012637 000004      MOV      (SP)+, @ERRVEC      ;; RESTORE THE ERROR VECTOR
9634 066250 000413      BR       7$                ;; LOOP ON THE PRESENT TEST
9635 066252      6$:     ; *****END OF CODE FOR THE XOR TESTER*****
9636 066252 105737 001103      2$:     TSTB     $ERFLG         ;; HAS AN ERROR OCCURRED?
9637 066256 001421      BEQ     3$                 ;; BR IF NO
9638 066260 123737 001115 001103      CMPB    $ERMAX, $ERFLG      ;; MAX. ERRORS FOR THIS TEST OCCURRED?
9639 066266 101015      BHI     3$                 ;; BR IF NO
9640 066270 032777 001000 112642      BIT     #BIT09, @SWR        ;; LOOP ON ERROR?
9641 066276 001404      BEQ     4$                 ;; BR IF NO
9642 066300 013737 001110 001106 7$:     MOV      $LPERR, $LPADR     ;; SET LOOP ADDRESS TO LAST SCOPE
9643 066306 000443      BR      $OVER              ;;
9644 066310 105037 001103      4$:     CLRB     $ERFLG         ;; ZERO THE ERROR FLAG
9645 066314 005037 001176      CLR     $TIMES             ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
9646 066320 000415      BR      1$                 ;; ESCAPE TO THE NEXT TEST
9647 066322 032777 004000 112610 3$:     BIT     #BIT11, @SWR        ;; INHIBIT ITERATIONS?
9648 066330 001011      BNE     1$                 ;; BR IF YES
9649 066332 005737 001100      TST     $PASS              ;; IF FIRST PASS OF PROGRAM
9650 066336 001406      BEQ     1$                 ;; INHIBIT ITERATIONS
9651 066340 005237 001104      INC     $ICNT              ;; INCREMENT ITERATION COUNT
9652 066344 033737 001176 001104      CMP     $TIMES, $ICNT      ;; CHECK THE NUMBER OF ITERATIONS MADE
9653 066352 002021      BGE     $OVER              ;; BR IF MORE ITERATION REQUIRED
9654 066354 012737 000001 001104 1$:     MOV      #1, $ICNT          ;; REINITIALIZE THE ITERATION COUNTER
9655 066362 013737 066432 001176      MOV     $MXCNT, $TIMES     ;; SET NUMBER OF ITERATIONS TO DO
9656 066370 105237 001102      $SVLAD: INCB     $STNM           ;; COUNT TEST NUMBERS
9657 066374 011637 001106      MOV     (SP), $LPADR       ;; SAVE SCOPE LOOP ADDRESS
9658 066400 011637 001110      MOV     (SP), $LPERR       ;; SAVE ERROR LOOP ADDRESS
9659 066404 005037 001200      CLR     $ESCAPE           ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
9660 066410 112737 000001 001115      MOVB   #1, $ERMAX         ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9661 066416 013777 001102 112516 $OVER:  MOV     $STNM, @DISPLAY    ;; DISPLAY TEST NUMBER
9662 066424 013716 001106      MOV     $LPADR, (SP)       ;; FUDGE RETURN ADDRESS
9663 066430 000002      RTI                       ;; FIXES PS
9664 066432 000004      $MXCNT: 4                 ;; MAX. NUMBER OF ITERATIONS
9665      .SBTTL  ERROR HANDLER ROUTINE
9666
9667      ; *****
9668      ; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9669      ; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9670      ; *AND GO TO $ERRTYP ON ERROR
9671      ; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9672      ; *SW15=1      HALT ON ERROR
9673      ; *SW13=1      INHIBIT ERROR TYPEOUTS
9674      ; *SW10=1     BELL ON ERROR
9675      ; *CALL
9676      ; *      ERROR  N      ;; ERROR=EMT AND N=ERROR ITEM NUMBER
9677
9678      $ERROR:
9679      066434 104407      CKSWR      ;; TEST FOR CHANGE IN SOFT-SWR
9680 066436 113737 001102 001246      MOVB   $STNM, $TSTNUM
9681 066444 105237 001103      7$:     INCB     $ERFLG         ;; SET THE ERROR FLAG
9682 066450 001775      BEQ     7$                 ;; DON'T LET THE FLAG GO TO ZERO
9683 066452 013777 001102 112462      MOV     $STNM, @DISPLAY    ;; DISPLAY TEST NUMBER AND ERROR FLAG
9684 066460 032777 002000 112452      BIT     #BIT10, @SWR        ;; BELL ON ERROR?
9685 066466 001402      BEQ     1$                 ;; NO - SKIP

```

ERROR HANDLER ROUTINE

```

9686 066470 104401 001202          TYPE      ,SBELL          ;;RING BELL
9687 066474 005237 001112      1$: INC      $ERTTL          ;;COUNT THE NUMBER OF ERRORS
9688 066500 011637 001116          MOV      (SP), $ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
9689 066504 162737 000002 001116          SUB      #2, $ERRPC
9690 066512 117737 112400 001114          MOV      @($ERRPC, $ITEMB) ;;STRIP AND SAVE THE ERROR ITEM CODE
9691 066520 032777 020000 112412          BIT      #BIT13, @SWR      ;;SKIP TYPEOUT IF SET
9692 066526 001004          BNE      20$              ;;SKIP TYPEOUTS
9693 066530 004737 066566          JSR      PC, $ERRRTP      ;;GO TO USER ERROR ROUTINE
9694 066534 104401 001207          TYPE      , $CRLF
9695 066540          20$:
9696 066540 005777 112374      2$: TST      @SWR              ;;HALT ON ERROR
9697 066544 100002          BPL      3$              ;;SKIP IF CONTINUE
9698 066546 000000          HALT
9699 066550 104407          CKSWR              ;;HALT ON ERROR!
9700 066552          3$:
9701 066552 022737 065726 000042          CMP      #SENDAD, @#42    ;;ACT-11 AUTO-ACCEPT?
9702 066560 001001          BNE      6$              ;;BRANCH IF NO
9703 066562 000000          HALT                  ;;YES
9704 066564          6$:
9705 066564 000002          RTI                  ;;RETURN
9706
9707 .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.

```

\$ERRRTP:

```

9713 066566          TYPE      , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
9714 066566 104401 001207          MOV      R0, -(SP)        ;; SAVE R0
9715 066572 010046          CLR      R0              ;; PICKUP THE ITEM INDEX
9716 066574 005000          BISB   @($ITEMB, R0
9717 066576 153700 001114          BNE      1$              ;; IF ITEM NUMBER IS ZERO, JUST
9718 066602 001004          1$:
9719          MOV      $ERRPC, -(SP) ;; TYPE THE PC OF THE ERROR
9720 066604 013746 001116          ;; SAVE $ERRPC FOR TYPEOUT
9721          ;; ERROR ADDRESS
9722 066610 104402          TYPCC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9723 066612 000445          JR      10$            ;; GET OUT
9724 066614 005300      1$: DEC      R0              ;; ADJUST THE INDEX SO THAT IT WILL
9725 066616 006300          ASL      R0              ;; WORK FOR THE ERROR TABLE
9726 066620 006300          ASL      R0
9727 066622 006300          ASL      R0
9728 066624 062700 001310          ADD      # $ERRTB, R0    ;; FORM TABLE POINTER
9729 066630 012037 066640          MOV      (R0)+, 2$      ;; PICKUP "ERROR MESSAGE" POINTER
9730 066634 001404          BEQ      3$              ;; SKIP TYPEOUT IF NO POINTER
9731 066636 104401          TYPE      "ERROR MESSAGE" ;; TYPE THE "ERROR MESSAGE"
9732 066640 000000      2$: .WORD   0              ;; "ERROR MESSAGE" POINTER GOES HERE
9733 066642 104401 001207          TYPE      , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
9734 066646 012037 066656      3$: MOV      (R0)+, 4$      ;; PICKUP "DATA HEADER" POINTER
9735 066652 001404          BEQ      5$              ;; SKIP TYPEOUT IF 0
9736 066654 104401          TYPE      "DATA HEADER"   ;; TYPE THE "DATA HEADER"
9737 066656 000000      4$: .WORD   0              ;; "DATA HEADER" POINTER GOES HERE
9738 066660 104401 001207          TYPE      , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
9739 066664 010146      5$: MOV      R1, -(SP)        ;; SAVE R1
9740 066666 012001          MOV      (R0)+, R1      ;; PICKUP "DATA TABLE" POINTER
9741 066670 001415          BEQ      9$              ;; BR IF NO DATA TO BE TYPED

```

```

9742 066672 012000      MOV      (R0)+,R0      ;; PICKUP "DATA FORMAT" POINTER
9743 066674 105720      6$: TSTB   (R0)+      ;; "OCTAL" OR "DECIMAL"
9744 066676 001003      BNE     7$           ;; BR IF DECIMAL
9745 066700 013146      MOV     @2(R1)+,-(SP) ;; SAVE @2(R1)+ FOR TYPEOUT
9746 066702 104402      TYPOC      ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9747 066704 000402      BR      8$
9748 066706      7$:
9749 066706 013146      MOV     @2(R1)+,-(SP) ;; SAVE @2(R1)+ FOR TYPEOUT
9750 066710 104405      TYPDS      ;; GO TYPE--DECIMAL ASCII WITH SIGN
9751 066712 005711      8$: TST     (R1)       ;; IS THERE ANOTHER NUMBER?
9752 066714 001403      BEQ     9$           ;; BR IF NO
9753 066716 104401 066736      TYPE     11$        ;; TYPE TWO(2) SPACES
9754 066722 000764      BR      6$           ;; LOOP
9755
9756 066724 012601      9$: MOV     (SP)+,R1    ;; RESTORE R1
9757 066726 012600      10$: MOV    (SP)+,R0   ;; RESTORE R0
9758 066730 104401 001207      TYPE     $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
9759 066734 000207      RTS     PC          ;; RETURN
9760 066736 020040 000      11$: .ASCIZ / /      ;; TWO(2) SPACES
9761      .EVEN
9762      .SBTTL TYPE ROUTINE
9763
9764      ;*****
9765      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
9766      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
9767      ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
9768      ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
9769      ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
9770      ;*
9771      ;*CALL:
9772      ;*1) USING A TRAP INSTRUCTION
9773      ;*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
9774      ;*OR
9775      ;*      TYPE
9776      ;*      MESADR
9777      ;*
9778
9779 066742 105737 001157      $TYPE: TSTB   $TFPLG      ;; IS THERE A TERMINAL?
9780 066746 100002      BPL     1$           ;; BR IF YES
9781 066750 000000      HALT      ;; HALT HERE IF NO TERMINAL
9782 066752 000407      BR      3$           ;; LEAVE
9783 066754 010046      1$: MOV     R0,-(SP)    ;; SAVE R0
9784 066756 017600 000002      MOV     @2(SP),R0    ;; GET ADDRESS OF ASCIZ STRING
9785 066762 112046      2$: MOV     (R0)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
9786 066764 001005      BNE     4$           ;; BR IF IT ISN'T THE TERMINATOR
9787 066766 005726      TST     (SP)+        ;; IF TERMINATOR POP IT OFF THE STACK
9788 066770 012600      60$: MOV    (SP)+,R0   ;; RESTORE R0
9789 066772 062716 000002      3$: ADD     #2,(SP)    ;; ADJUST RETURN PC
9790 066776 000002      RTI      ;; RETURN
9791 067000 122716 000011      4$: CMPB   #HT,(SP)    ;; BRANCH IF <HT>
9792 067004 001430      BEQ     8$
9793 067006 122716 000200      CMPB   #CRLF,(SP)   ;; BRANCH IF NOT <CRLF>
9794 067012 001006      BNE     5$
9795 067014 005726      TST     (SP)+        ;; POP <CR><LF> EQUIV
9796 067016 104401      TYPE     ;; TYPE A CR AND LF
9797 067020 001207      $CRLF

```

```

9798 067022 105037 067156 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
9799 067026 000755 BR 2$ ;; GET NEXT CHARACTER
9800 067030 004737 067112 5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
9801 067034 123726 001156 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
9802 067040 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
9803 067042 013746 001154 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
9804 ;; AND THE NULL CHAR.
9805 067046 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
9806 067052 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
9807 067054 004737 067112 JSR PC,$TYPEC ;; GO TYPE A NULL
9808 067060 105337 067156 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
9809 067064 000770 BR 7$ ;; LOOP

```

;HORIZONTAL TAB PROCESSOR

```

9813 067066 112716 000040 8$: MOVB #' (SP) ;; REPLACE TAB WITH SPACE
9814 067072 004737 067112 9$: JSR PC,$TYPEC ;; TYPE A SPACE
9815 067076 132737 000007 067156 BITB #',$CHARCNT ;; BRANCH IF NOT AT
9816 067104 001372 BNE 9$ ;; TAB STOP
9817 067106 005726 TST (SP)+ ;; POP SPACE OFF STACK
9818 067110 000724 BR 2$ ;; GET NEXT CHARACTER
9819 067112 105777 112032 $TYPEC: TSTB 2$TPS ;; WAIT UNTIL PRINTER IS READY
9820 067116 000375 BPL $TYPEC
9821 067120 116677 000002 112024 MOVB 2(SP),2$TPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.
9822 067126 122766 000015 000002 CMPB #CR,2(SP) ;; IS CHARACTER A CARRIAGE RETURN?
9823 067134 001003 BNE 1$ ;; BRANCH IF NO
9824 067136 105037 067156 CLRB $CHARCNT ;; YES--CLEAR CHARACTER COUNT
9825 067142 000406 BR $TYPEX ;; EXIT
9826 067144 122766 000012 000002 1$: CMPB #LF,2(SP) ;; IS CHARACTER A LINE FEED?
9827 067152 001402 BEQ $TYPEX ;; BRANCH IF YES
9828 067154 105227 INCB (PC)+ ;; COUNT THE CHARACTER
9829 067156 000000 $CHARCNT: WORD 0 ;; CHARACTER COUNT STORAGE
9830 067160 000207 $TYPEX: RTS PC

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

9831
9832 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
9833
9834 *****
9835 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
9836 *OCTAL (ASCII) NUMBER AND TYPE IT.
9837 *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
9838 *CALL:
9839 * MOV NUM,-(SP) ;; NUMBER TO BE TYPED
9840 * TYPOS ;; CALL FOR TYPEOUT
9841 * .BYTE N ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
9842 * .BYTE M ;; M=1 OR 0
9843 * ;; 1=TYPE LEADING ZEROS
9844 * ;; 0=SUPPRESS LEADING ZEROS
9845
9846 *$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
9847 *$TYPOS OR $TYPOC
9848 *CALL:
9849 * MOV NUM,-(SP) ;; NUMBER TO BE TYPED
9850 * TYPON ;; CALL FOR TYPEOUT
9851
9852 *$STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
9853 *CALL:

```

M14

MO-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 181
 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0183

```

9854          ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
9855          ;*      TYPOC      ;;CALL FOR TYPEOUT
9856
9857 067162 017646 000000          $TYPOS: MOV      2(SP),-(SP)      ;; PICKUP THE MODE
9858 067166 116637 000001 067405  MOVB     1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
9859 067174 112637 067407          MOVB     (SP)+,SOMODE+1    ;;NUMBER OF DIGITS TO TYPE
9860 067200 062716 000002          ADD      #2,(SP)        ;;ADJUST RETURN ADDRESS
9861 067204 000406          BR       $TYPON
9862 067206 112737 000001 067405  $TYPOC: MOVB     #1,SOFILL      ;;SET THE ZERO FILL SWITCH
9863 067214 112737 000006 067407  MOVB     #6,SOMODE+1    ;;SET FOR SIX(6) DIGITS
9864 067222 112737 000005 067404  $TYPON: MOVB     #5,SOCNT      ;;SET THE ITERATION COUNT
9865 067230 010346          MOV      R3,-(SP)      ;;SAVE R3
9866 067232 010446          MOV      R4,-(SP)      ;;SAVE R4
9867 067234 010546          MOV      R5,-(SP)      ;;SAVE R5
9868 067236 113704 067407          MOVB     $SOMODE+1,R4   ;;GET THE NUMBER OF DIGITS TO TYPE
9869 067242 005404          NEG      R4
9870 067244 062704 000006          ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
9871 067250 110437 067406          MOVB     R4,SOMODE      ;;SAVE IT FOR USE
9872 067254 113704 067405          MOVB     $SOFILL,R4     ;;GET THE ZERO FILL SWITCH
9873 067260 016605 000012          MOV      12(SP),R5     ;;PICKUP THE INPUT NUMBER
9874 067264 005003          CLR      R3            ;;CLEAR THE OUTPUT WORD
9875 067266 006105          1$:     ROL      R5          ;;ROTATE MSB INTO "C"
9876 067270 000404          BR       3$
9877 067272 006105          2$:     ROL      R5          ;;GO DO MSB
9878 067274 006105          ROL      R5            ;;FORM THIS DIGIT
9879 067276 006105          ROL      R5
9880 067300 010503          MOV      R5,R3
9881 067302 006103          3$:     ROL      R3          ;;GET LSB OF THIS DIGIT
9882 067304 105337 067406          DECB     $SOMODE        ;;TYPE THIS DIGIT?
9883 067310 107016          BPL      7$
9884 067312 042703 177770          BIC      #177770,R3     ;;BR IF NO
9885 067316 001002          BNE      4$            ;;GET RID OF JUNK
9886 067320 005704          TST     R4            ;;TEST FOR 0
9887 067322 001403          BEQ     5$            ;;SUPPRESS THIS 0?
9888 067324 005204          4$:     INC     R4          ;;BR IF YES
9889 067326 052703 000060          BIS     #'0,R3        ;;DON'T SUPPRESS ANYMORE 0'S
9890 067332 052703 000040          5$:     BIS     #' ,R3     ;;MAKE THIS DIGIT ASCII
9891 067336 110337 067402          MOVB     R3,8$         ;;MAKE ASCII IF NOT ALREADY
9892 067342 104401 067402          TYPE     8$           ;;SAVE FOR TYPING
9893 067346 105337 067404          7$:     DECB     $SOCNT     ;;GO TYPE THIS DIGIT
9894 067352 003347          BGT     2$            ;;COUNT BY 1
9895 067354 002402          BLT     6$            ;;BR IF MORE TO DO
9896 067356 005204          INC     R4            ;;BR IF DONE
9897 067360 000744          BR      2$           ;;INSURE LAST DIGIT ISN'T A BLANK
9898 067362 012605          6$:     MOV     (SP)+,R5     ;;GO DO THE LAST DIGIT
9899 067364 012604          MOV     (SP)+,R4       ;;RESTORE R5
9900 067366 012603          MOV     (SP)+,R3       ;;RESTORE R4
9901 067370 016666 000002 000004  MOV     2(SP),4(SP)    ;;RESTORE R3
9902 067376 012616          MOV     (SP)+,(SP)    ;;SET THE STACK FOR RETURNING
9903 067400 000002          RTI
9904 067402 000          8$:     .BYTE 0          ;;RETURN
9905 067403 000          .BYTE 0          ;;STORAGE FOR ASCII DIGIT
9906 067404 000          $SOCNT: .BYTE 0     ;;TERMINATOR FOR TYPE ROUTINE
9907 067405 000          $SOFILL: .BYTE 0    ;;OCTAL DIGIT COUNTER
9908 067406 000000          $SOMODE: .WORD 0    ;;ZERO FILL SWITCH
9909          .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
  
```

```

9910
9911      ;*****
9912      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
9913      ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
9914      ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
9915      ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
9916      ;*REPLACED WITH SPACES.
9917      ;*CALL:
9918      ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
9919      ;*      TYPDS      ;;GO TO THE ROUTINE
9920
9921      $TYPDS:
9922      067410      010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
9923      067412      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9924      067414      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
9925      067416      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
9926      067420      010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
9927      067422      012746      020200      MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
9928      067426      016605      000020      MOV      20(SP),R5      ;;GET THE INPUT NUMBER
9929      067432      100004      BPL      1$      ;;BR IF INPUT IS POS.
9930      067434      005405      NEG      R5      ;;MAKE THE BINARY NUMBER POS.
9931      067436      112766      000055      000001      MOV      #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
9932      067444      005000      1$:      CLR      R0      ;;ZERO THE CONSTANTS INDEX
9933      067446      0127C3      067624      MOV      #$DBLK,R3      ;;SETUP THE OUTPUT POINTER
9934      067452      112723      000040      MOV      #' ,(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
9935      067456      005002      2$:      CLR      R2      ;;CLEAR THE BCD NUMBER
9936      067460      016001      067614      MOV      $DTBL(R0),R1      ;;GET THE CONSTANT
9937      067464      160105      3$:      SUB      R1,R5      ;;FORM THIS BCD DIGIT
9938      067466      002402      BLT      4$      ;;BR IF DONE
9939      067470      005202      INC      R2      ;;INCREASE THE BCD DIGIT BY 1
9940      067472      000774      BR      3$
9941      067474      060105      4$:      ADD      R1,R5      ;;ADD BACK THE CONSTANT
9942      067476      005702      TST      R2      ;;CHECK IF BCD DIGIT=0
9943      067500      001002      BNE      5$      ;;FALL THROUGH IF 0
9944      067502      105716      TSTB     (SP)      ;;STILL DOING LEADING 0'S?
9945      067504      100407      BMI      7$      ;;BR IF YES
9946      067506      106316      5$:      ASLB     (SP)      ;;MSD?
9947      067510      103003      BCC      6$      ;;BR IF NO
9948      067512      116663      000001      177777      MOV      1(SP),-1(R3)      ;;YES--SET THE SIGN
9949      067520      052702      00006C      6$:      BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII
9950      067524      052702      000040      7$:      BIS      #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9951      067530      110223      MOV      R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9952      067532      005720      TST      (R0)+      ;;JUST INCREMENTING
9953      067534      020027      000010      CMP      R0,#10      ;;CHECK THE TABLE INDEX
9954      067540      002746      BLT      2$      ;;GO DO THE NEXT DIGIT
9955      067542      003002      BGT      8$      ;;GO TO EXIT
9956      067544      010502      MOV      R5,R2      ;;GET THE LSD
9957      067546      000764      BR      6$      ;;GO CHANGE TO ASCII
9958      067550      105726      8$:      TSTB     (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
9959      067552      100003      BPL      9$      ;;BR IF NO
9960      067554      116663      177777      177776      MOV      -(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
9961      067562      105013      9$:      CLRB     (R3)      ;;SET THE TERMINATOR
9962      067564      012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
9963      067566      012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
9964      067570      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
9965      067572      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1

```



```

9966 067574 012600          MOV      (SP)+,RO      ;; POP STACK INTO RO
9967 067576 104401 067624    TYPE      $DBLK        ;; NOW TYPE THE NUMBER
9968 067602 016666 000002 000004    MOV      2(SP),4(SP)  ;; ADJUST THE STACK
9969 067610 012616          MOV      (SP)+,(SP)
9970 067612 000002          RTI                    ;; RETURN TO USER
9971 067614 023420          $DTBL: 10000.
9972 067616 001750          1000.
9973 067620 000144          100.
9974 067622 000012          10.
9975 067624 000004          $DBLK: .BLKW 4
9976                                .SBTTL TTY INPUT ROUTINE
9977
9978                                ;;*****
9979                                .ENABL LSB
9980 067634 000000          $TKCNT: .WORD 0      ;; NUMBER OF ITEMS IN QUEUE
9981 067636 000000          $TKQIN: .WORD 0     ;; INPUT POINTER
9982 067640 000000          $TKQOUT: .WORD 0    ;; OUTPUT POINTER
9983 067642 000001          $TKQSRV: .BLKB 1   ;; TTY KEYBOARD QUEUE
9984                                $TKQEND=.
9985                                .EVEN
9986
9987                                ;*TK INITIALIZE ROUTINE
9988                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
9989                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
9990
9991                                ;*CALL:
9992                                ;*      JSR      PC,$TKINT
9993                                ;*      RETURN
9994
9995 067644 005037 067634          $TKINT: CLR      $TKCNT      ;; CLEAR COUNT OF ITEMS IN QUEUE
9996 067650 012737 067642 067636    MOV      $TKQSRV,$TKQIN ;; MOVE THE STARTING ADDRESS OF THE
9997 067656 013737 067636 067640    MOV      $TKQIN,$TKQOUT ;; QUEUE INTO THE INPUT & OUTPUT PC TERS.
9998 067664 012737 067714 000060    MOV      $TKSRV,$TKVEC ;; INITIALIZE THE KEYBOARD VECTOR
9999 067672 012737 000200 000062    MOV      #200,$TKVEC+2 ;; "BR" LEVEL 4
10000 067700 005777 111242          TST      $TKB         ;; CLEAR DONE FLAG
10001 067704 012777 000100 111232    MOV      #100,$TKS    ;; ENABLE TTY KEYBOARD INTERRUPT
10002 067712 000207          RTS      PC          ;; RETURN TO CALLER
10003
10004                                ;*TK SERVICE ROUTINE
10005                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
10006                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
10007                                ;*IT IN THE QUEUE.
10008
10009 067714 117746 111226          $TKSRV: MOVSB  $TKB,-(SP) ;; PICK UP THE CHARACTER
10010 067720 042716 177600          BIC      #177,(SP)    ;; STRIP THE JUNK
10011 067724 021627 000007          1$:     CMP      (SP),#? ;; IS IT A CONTROL G?
10012 067730 001004          BNE      2$          ;; BRANCH IF NO
10013 067732 022737 000176 001140    CMP      $SWREG,$SWR  ;; IS SOFT-SWR SELECTED?
10014 067740 001500          BEQ      6$          ;; GO TO SWR CHANGE
10015
10016                                2$:
10017 067742 022737 000001 067634          CMP      #1,$TKCNT   ;; IS THE QUEUE FULL?
10018 067750 001004          BNE      3$          ;; BRANCH IF NO
10019 067752 104401 001202          TYPE      $BELL     ;; RING THE TTY BELL
10020 067756 005726          TST      (SP)+      ;; CLEAN CHARACTER OFF OF STACK
10021 067760 000451          BR      5$          ;; EXIT

```

TTY INPUT ROUTINE

```

10022 067762 021627 000023 3$: CMP (SP),#23 ;: IS IT A CONTROL-S?
10023 067766 001021 BNE 32$ ;: BRANCH IF NO
10024 067770 005077 111150 CLR @STKS ;: DISABLE TTY KEYBOARD INTERRUPTS
10025 067774 005726 TST (SP)+ ;: CLEAN CHAR OFF STACK
10026 067776 105777 111142 31$: TSTB @STKS ;: WAIT FOR A CHAR
10027 070002 100375 BPL 31$ ;: LOOP UNTIL ITS THERE
10028 070004 117746 111136 MOVB @STKB,-(SP) ;: GET THE CHARACTER
10029 070010 042716 177600 BIC #177,(SP) ;: MAKE IT 7-BIT ASCII
10030 070014 022627 000021 CMP (SP)+,#21 ;: IS IT A CONTROL-Q?
10031 070020 001366 BNE 31$ ;: BRANCH IF NO
10032 070022 012777 000100 111114 MOV #100,@STKS ;: REENABLE TTY KEYBOARD INTERRUPTS
10033 070030 000002 RTI ;: RETURN
10034 070032 005237 067634 32$: INC $TKCNT ;: COUNT THIS CHARACTER
10035 070036 021627 000140 CMP (SP),#140 ;: IS IT UPPER CASE?
10036 070042 002405 BLT 4$ ;: BRANCH IF YES
10037 070044 021627 000175 CMP (SP),#175 ;: IS IT A SPECIAL CHAR?
10038 070050 003002 BGT 4$ ;: BRANCH IF YES
10039 070052 042716 000040 BIC #40,(SP) ;: MAKE IT UPPER CASE
10040 070056 112677 177554 4$: MOVB (SP)+,@STKQIN ;: AND PUT IT IN QUEUE
10041 070062 005237 067636 INC $TKQIN ;: UPDATE THE POINTER
10042 070066 023727 067636 067643 CMP $TKQIN,$STKQEND ;: GO OFF THE END?
10043 070074 001003 BNE 5$ ;: BRANCH IF NO
10044 070076 012737 067642 067636 5$: MOV #STKQSR,$STKQIN ;: RESET THE POINTER
10045 070104 000002 RTI ;: RETURN

```

```

10046
10047 ;: *****
10048 ;: *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
10049 ;: *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
10050 ;: *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
10051 ;: *CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

10052 070106 022737 000176 001140 $CKSWR: CMP #SWREG,$SWR ;: IS THE SOFT-SWR SELECTED
10053 070114 001104 BNE 15$ ;: EXIT IF NOT
10054 070116 105777 111022 TSTB @STKS ;: IS A CHAR WAITING?
10055 070122 100101 BPL 15$ ;: IF NOT, EXIT
10056 070124 117746 111016 MOVB @STKB,-(SP) ;: YES
10057 070130 042716 177600 BIC #177,(SP) ;: MAKE IT 7-BIT ASCII
10058 070134 021627 000007 CMP (SP),#7 ;: IS IT A CONTROL-G?
10059 070140 001300 BNE 2$ ;: IF NOT, PUT IT IN THE TTY QUEUE
10060 ;: AND EXIT

```

```

10061
10062 ;: *****
10063 ;: *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
10064 ;: *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
10065 ;: *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

10066 070142 123727 001134 000001 6$: CMPB $AUTOB,#1 ;: ARE WE RUNNING IN AUTO-MODE?
10067 070150 001674 BEQ 2$ ;: BRANCH IF YES
10068 070152 005726 TST (SP)+ ;: CLEAR CONTROL-G OFF STACK
10069 070154 004737 067644 TSR PC,$TKINT ;: FLUSH THE TTY INPUT QUEUE
10070 070160 005077 110760 CLR @STKS ;: DISABLE TTY KEYBOARD INTERRUPTS
10071 070164 112737 000001 001135 MOVB #1,$INTAG ;: SET INTERRUPT MODE INDICATOR
10072
10073 070172 104401 070750 TYPE , $CNTLG ;: ECHO THE CONTROL-G (+G)
10074 070176 104401 070755 $GTSWR: TYPE , $MSWR ;: TYPE CURRENT CONTENTS
10075 070202 013746 000176 MOV $SWREG,-(SP) ;: SAVE SWREG FOR TYPEOUT
10076 070206 104402 TYPOC ;: GO TYPE--OCTAL ASCII(ALL DIGITS)
10077 070210 104401 070766 TYPE , MNEW ;: PROMPT FOR NEW SWR

```

TTY INPUT ROUTINE

```

10078 070214 005046          19$: CLR      -(SP)          ;; CLEAR COUNTER
10079 070216 005046          CLR      -(SP)          ;; THE NEW SWR
10080 070220 105777 110720  7$:  TSTB   2$TKS        ;; CHAR THERE?
10081 070224 100375          BPL      7$            ;; IF NOT TRY AGAIN
10082
10083 070226 117746 110714   MOVB   2$TKB,-(SP)      ;; PICK UP CHAR
10084 070232 042716 177600   BIC     #1C177,(SP)    ;; MAKE IT 7-BIT ASCII
10085
10086
10087
10088 070236 021627 000025   9$:  CMP     (SP),#25    ;; IS IT A CONTROL-U?
10089 070242 001005          BNE     10$           ;; BRANCH IF NOT
10090 070244 104401 070743   TYPE   $CNTLU        ;; YES, ECHO CONTROL-U (↑U)
10091 070250 062706 000006  20$:  ADD     #6,SP      ;; IGNORE PREVIOUS INPUT
10092 070254 000757          BR      19$          ;; LET'S TRY IT AGAIN
10093
10094
10095 070256 021627 000015   10$:  CMP     (SP),#15   ;; IS IT A <CR>?
10096 070262 001022          BNE     16$           ;; BRANCH IF NO
10097 070264 005766 000004   TST     4(SP)        ;; YES, IS IT THE FIRST CHAR?
10098 070270 001403          BEQ     11$           ;; BRANCH IF YES
10099 070272 016677 000002 110640  MOV     2(SP),2$SWR   ;; SAVE NEW SWR
10100 070300 062706 000006  11$:  ADD     #6,SP      ;; CLEAR UP STACK
10101 070304 104401 001207  14$:  TYPE   $CRLF        ;; ECHO <CR> AND <LF>
10102 070310 123727 001135 000001  CMPB   $INTAG,#1     ;; RE-ENABLE TTY KBD INTERRUPTS?
10103 070316 001003          BNE     15$           ;; BRANCH IF NOT
10104 070320 012777 000100 110616  MOV     #100,2$TKS   ;; RE-ENABLE TTY KBD INTERRUPTS
10105 070326 000002          RTI                    ;; RETURN
10106 070330 004737 067112  16$:  JSR     PC,$TYPEC    ;; ECHO CHAR
10107 070334 021627 000060   CMP     (SP),#60     ;; CHAR < 0?
10108 070340 002420          BLT     18$           ;; BRANCH IF YES
10109 070342 021627 000067   CMP     (SP),#67     ;; CHAR > 7?
10110 070346 003015          BGT     18$           ;; BRANCH IF YES
10111 070350 042726 000060   BIC     #60,(SP)+    ;; STRIP-OFF ASCII
10112 070354 005766 000002   TST     2(SP)        ;; IS THIS THE FIRST CHAR
10113 070360 001403          BEQ     17$           ;; BRANCH IF YES
10114 070362 006316          ASL     (SP)         ;; NO, SHIFT PRESENT
10115 070364 006316          ASL     (SP)         ;; CHAR OVER TO MAKE
10116 070366 006316          ASL     (SP)         ;; ROOM FOR NEW ONE.
10117 070370 005266 000002  17$:  INC     2(SP)        ;; KEEP COUNT OF CHAR
10118 070374 056616 177776   BIS     -2(SP),(SP)  ;; SET IN NEW CHAR
10119 070400 000707          BR      7$            ;; GET THE NEXT ONE
10120 070402 104401 001206  18$:  TYPE   $QUES        ;; TYPE ?<CR><LF>
10121 070406 000720          BR      20$          ;; SIMULATE CONTROL-U
10122 .DSABL  LSB
10123
10124
10125 *****
10126 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10127 *CALL:
10128 *   RDCHR          ;; GET A CHARACTER FROM THE QUEUE
10129 *   RETURN HERE   ;; CHARACTER IS ON THE STACK
10130 *               ;; WITH PARITY BIT STRIPPED OFF
10131
10132
10133 070410 011646  $RDCHR: MOV     (SP),-(SP) ;; PUSH DOWN THE PC AND

```

```

10134 070412 016666 000004 000002      MOV    4(SP),2(SP)      ;; THE PS
10135 070420 005066 000004          CLR    4(SP)           ;; GET READY FOR A CHARACTER
10136 070424 005046          CLR    -(SP)          ;; PUT NEW PS ON STACK
10137 070426 012746 070434      MOV    #64$,-(SP)     ;; PUT NEW PC ON STACK
10138 070432 000002          RTI                   ;; POP NEW PC AND PS
10139 070434          64$:
10140 070434 005737 067634      1$:    TST    STKCNT      ;; WAIT ON A CHARACTER
10141 070440 001775          BEQ    1$             ;;
10142 070442 005337 067634      DEC    STKCNT         ;; DECREMENT THE COUNTER
10143 070446 117766 177166 000004      MOVB  STKQOUT,4(SP)   ;; GET ONE CHARACTER
10144 070454 005237 067640      INC    STKQOUT        ;; UPDATE THE POINTER
10145 070460 023727 067640 067643      CMP    STKQOUT,#STKQEND ;; DID IT GO OFF OF THE END?
10146 070466 001003          BNE    2$            ;; BRANCH IF NO
10147 070470 012737 067642 067640      MOV    #STKQSRST,STKQOUT ;; RESET THE POINTER
10148 070476 000002          RTI                   ;; RETURN
10149          ;; *****
10150          ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10151          ;; *CALL:
10152          ;; *
10153          ;; *   RDLIN          ;; INPUT A STRING FROM THE TTY
10154          ;; *   RETURN HERE  ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10155          ;; *
10156          ;; *
10156 070500 010346      $RDLIN: MOV    R3, -(SP)     ;; SAVE R3
10157 070502 005046          CLR    -(SP)         ;; CLEAR THE RUBOUT KEY
10158 070504 012703 070734      1$:    MOV    #STTYIN,R3 ;; GET ADDRESS
10159 070510 022703 070743      2$:    CMP    #STTYIN+7,R3 ;; BUFFER FULL?
10160 070514 101456          BLOS   4$            ;; BR IF YES
10161 070516 104410          RDCHR          ;; GO READ ONE CHARACTER FROM THE TTY
10162 070520 112613          MOVB  (SP)+(R3)      ;; GET CHARACTER
10163 070522 122713 000177      10$:   CMPB  #177,(R3)     ;; IS IT A RUBOUT
10164 070526 001022          BNE    5$            ;; BR IF NO
10165 070530 005716          TST   (SP)          ;; IS THIS THE FIRST RUBOUT?
10166 070532 001007          BNE    6$            ;; BR IF NO
10167 070534 112737 000134 070732      MOVB  #'\\,9$        ;; TYPE A BACK SLASH
10168 070542 104401 070732          TYPE  9$
10169 070546 012716 177777          MOV    #-1,(SP)     ;; SET THE RUBOUT KEY
10170 070552 005303          DEC   R3             ;; BACKUP BY ONE
10171 070554 020327 070734      6$:    CMP    R3,#STTYIN  ;; STACK EMPTY?
10172 070560 103434          BLO   4$            ;; BR IF YES
10173 070562 111337 070732          MOVB  (R3),9$       ;; SETUP TO TYPEOUT THE DELETED CHAR.
10174 070566 104401 070732          TYPE  9$
10175 070572 000746          BR    2$            ;; GO READ ANOTHER CHAR.
10176 070574 005716          5$:    TST   (SP)          ;; RUBOUT KEY SET?
10177 070576 001406          BEQ   7$            ;; BR IF NO
10178 070600 112737 000134 070732      MOVB  #'\\,9$        ;; TYPE A BACK SLASH
10179 070606 104401 070732          TYPE  9$
10180 070612 005016          CLR   (SP)          ;; CLEAR THE RUBOUT KEY
10181 070614 122713 000025      7$:    CMPB  #25,(R3)     ;; IS CHARACTER A CTRL U?
10182 070620 001003          BNE   8$            ;; BR IF NO
10183 070622 104401 070743          TYPE  %CNTLU        ;; TYPE A CONTROL "U"
10184 070626 000726          BR    1$            ;; GO START OVER
10185 070630 122713 000022      8$:    CMPB  #22,(R3)     ;; IS CHARACTER A "↑R"?
10186 070634 001011          BNE   3$            ;; BRANCH IF NO
10187 070636 105013          CLRB  (R3)          ;; CLEAR THE CHARACTER
10188 070640 104401 001207          TYPE  %CRLF        ;; TYPE A "CR" & "LF"
10189 070644 104401 070734          TYPE  %STTYIN      ;; TYPE THE INPUT STRING

```

```

10190 070650 000717          BR      2$          ;; GO PICKUP ANOTHER CHACTER
10191 070652 104401 001206 4$:  TYPE      $QUES      ;; TYPE A '?'
10192 070656 000712          BR      1$          ;; CLEAR THE BUFFER AND LOOP
10193 070660 111337 070732 3$:  MOVB      (R3),9$      ;; ECHO THE CHARACTER
10194 070664 104401 070732          TYPE      ,9$
10195 070670 122723 000015          CMPB     #15,(R3)+      ;; CHECK FOR RETURN
10196 070674 011305          BNE      2$          ;; LOOP IF NOT RETURN
10197 070676 105063 177777          CLRB     -1(R3)       ;; CLEAR RETURN (THE 15)
10198 070702 104401 001210          TYPE     $LF         ;; TYPE A LINE FEED
10199 070706 005726          TST      (SP)+       ;; CLEAN RUBOUT KEY FROM THE STACK
10200 070710 012603          MOV      (SP)+,R3    ;; RESTORE R3
10201 070712 011646          MOV      (SP)-,(SP)  ;; ADJUST THE STACK AND PUT ADDRESS OF THE
10202 070714 016666 000004 000002 MOV      4(SP),2(SP)  ;; FIRST ASCII CHARACTER ON IT
10203 070722 012766 070734 000004 MOV      #STTYIN,4(SP)
10204 070730 007002          RTI
10205 070732 000          9$:  .BYTE     0          ;; STORAGE FOR ASCII CHAR. TO TYPE
10206 070733 000          .BYTE     0          ;; TERMINATOR
10207 070734 000007          $TTYIN: .BLKB     7          ;; RESERVE 7 BYTES FOR TTY INPUT
10208 070743 136 006525 000012 $CNTLU: .ASCIZ  /↑U<15><12>      ;; CONTROL "U"
10209 070750 043536 005015 000 $CNTLG: .ASCIZ  /↑G<15><12>      ;; CONTROL "G"
10210 070755 015 051412 051127 $MSWR:  .ASCIZ  <15><12>/SWR = /
10211 070762 036440 000040          $MNEW:  .ASCIZ  / NEW = /
10212 070766 020040 042516 020127 .EVEN
10213 070774 020075 000          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
10214 071000
10215
10216
10217
10218
10219
10220
10221
10222
10223
10224
10225
10226
10227
10228
10229 071000 011646          ;; *****
10230 071002 016666 000004 000002 ;; *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
10231 071010 010046          ;; *CHANGE IT TO BINARY.
10232 071012 010146          ;; *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
10233 071014 010246          ;; *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
10234 071016 104411          ;; *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
10235 071020 012600          ;; *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
10236 071022 010037 071126          ;; *CALL:
10237 071026 005001          ;; *
10238 071030 005002          ;; * RDOCT          ;; READ AN OCTAL NUMBER
10239 071032 112046          ;; *          ;; LOW ORDER BITS ARE ON TOP OF THE STACK
10240 071034 001420          ;; * RETURN HERE          ;; HIGH ORDER BITS ARE IN $HIOCT
10241 071036 122716 000060 $RDOCT: MOV      (SP)-,(SP)      ;; PROVIDE SPACE FOR THE
10242 071042 003026          MOV      4(SP),2(SP)      ;; INPUT NUMBER
10243 071044 122716 000067          MOV      R0,-(SP)        ;; PUSH R0 ON STACK
10244 071050 002423          MOV      R1,-(SP)        ;; PUSH R1 ON STACK
10245 071052 006301          MOV      R2,-(SP)        ;; PUSH R2 ON STACK
          1$:  RDLIN          ;; READ AN ASCIZ LINE
          MOV      (SP)+,R0      ;; GET ADDRESS OF 1ST CHARACTER
          MOV      R0,5$        ;; AND SAVE IT
          CLR      R1          ;; CLEAR DATA WORD
          CLR      R2
          2$:  MOVB      (R0)+,-(SP)  ;; PICKUP THIS CHARACTER
          BEQ      3$          ;; IF ZERO GET OUT
          CMPB     #'0,(SP)    ;; MAKE SURE THIS CHARACTER
          BGT      4$          ;; IS AN OCTAL DIGIT
          CMPB     #'7,(SP)
          BLT      4$
          ASL      R1          ;; *2

```

```

10246 071054 006102          ROL      R2
10247 071056 006301          ASL      R1          ;;*4
10248 071060 006102          ROL      R2
10249 071062 006301          ASL      R1          ;;*8
10250 071064 006102          ROL      R2
10251 071066 042716 177770  BIC      #1C7,(SP)   ;;STRIP THE ASCII JUNK
10252 071072 062601          ADD      (SP)+,R1   ;;ADD IN THIS DIGIT
10253 071074 000756          BR       2$        ;;LOOP
10254 071076 005726          3$:     TST      (SP)+   ;;CLEAN TERMINATOR FROM STACK
10255 071100 010166 000012     MOV      R1,12(SP) ;;SAVE THE RESULT
10256 071104 010237 071136     MOV      R2,$SHIOCT
10257 071110 012602          MOV      (SP)+,R2   ;;POP STACK INTO R2
10258 071112 012601          MOV      (SP)+,R1   ;;POP STACK INTO R1
10259 071114 012600          MOV      (SP)+,R0   ;;POP STACK INTO R0
10260 071116 000002          RTI
10261 071120 005726          4$:     TST      (SP)+   ;;CLEAN PARTIAL FROM STACK
10262 071122 105010          CLR      (R0)      ;;SET A TERMINATOR
10263 071124 104401          TYPE
10264 071126 000000          5$:     .WORD    0      ;;TYPE UP THRU THE BAD CHAR.
10265 071130 104401 001206     TYPE    $QUES      ;;"? "CR" & "LF"
10266 071134 000730          BR       1$        ;;TRY AGAIN
10267 071136 000000          $SHIOCT: .WORD    0   ;;HIGH ORDER BITS GO HERE
10268
10269          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
10270
10271          ;*****
10272          ;*SAVE R0-R5
10273          ;*CALL:
10274          ;*   SAVREG
10275          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10276          ;*
10277          ;*TOP---(+16)
10278          ;* +2---(+18)
10279          ;* +4---R5
10280          ;* +6---R4
10281          ;* +8---R3
10282          ;*+10---R2
10283          ;*+12---R1
10284          ;*+14---R0
10285
10286          $SAVREG:
10287          MOV      R0,-(SP)   ;;PUSH R0 ON STACK
10288          MOV      R1,-(SP)   ;;PUSH R1 ON STACK
10289          MOV      R2,-(SP)   ;;PUSH R2 ON STACK
10290          MOV      R3,-(SP)   ;;PUSH R3 ON STACK
10291          MOV      R4,-(SP)   ;;PUSH R4 ON STACK
10292          MOV      R5,-(SP)   ;;PUSH R5 ON STACK
10293          MOV      22(SP),-(SP) ;;SAVE PS OF MAIN FLOW
10294          MOV      22(SP),-(SP) ;;SAVE PC OF MAIN FLOW
10295          MOV      22(SP),-(SP) ;;SAVE PC OF CALL
10296          MOV      22(SP),-(SP) ;;SAVE PC OF CALL
10297          RTI
10298
10299          ;*RESTORE R0-R5
10300          ;*CALL:
10301          ;*   RESREG
          $RESREG:

```

10302 071176 012666 000022
 10303 071202 012666 000022
 10304 071206 012666 000022
 10305 071212 012666 000022
 10306 071216 012605
 10307 071220 012604
 10308 071222 012603
 10309 071224 012602
 10310 071226 012601
 10311 071230 012600
 10312 071232 000002

```

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

```

.SBTTL TRAP DECODER

10313
 10314
 10315
 10316
 10317
 10318
 10319
 10320

```

*****
;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.

```

10321 071234 010046
 10322 071236 016600 000002
 10323 071242 005740
 10324 071244 111000
 10325 071246 006300
 10326 071250 016000 071270
 10327 071254 000200

```

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

```

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

10331
 10332 071256 011646
 10333 071260 016666 000004 000002
 10334 071266 000002
 10335

```

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

```

.SBTTL TRAP TABLE

10336
 10337
 10338
 10339
 10340
 10341
 10342

```

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

```

10343 071270 071256
 10344 071272 066742
 10345 071274 067206
 10346 071276 067162
 10347 071300 067222
 10348 071302 067410
 10349
 10350 071304 070176
 10351
 10352 071306 070106
 10353 071310 070410
 10354 071312 070500
 10355 071314 071000
 10356 071316 071140
 10357 071320 071176

```

; ROUTINE
;-----
$TRPAD: .WORD $TRAP2
$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
$TYPC ;;CALL=TYPC 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)

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

$CKSWR ;;CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
$RDCHR ;;CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
$RDLIN ;;CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
$RDOCT ;;CALL=RDOCT TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
$SAVREG ;;CALL=SAVREG TRAP+13(104413) SAVE RO-R5 ROUTINE
$RESREG ;;CALL=RESREG TRAP+14(104414) RESTORE RO-R5 ROUTINE

```

10358				
10359				
10360				
10361				
10362				
10363				
10364				
10365	071322	005015	046412	044501
10366	071330	042116	041505	030455
10367	071336	026461	055104	045122
10368	071344	026505	006501	012
10369	071351	122	030115	020063
10370	071356	052504	046101	041440
10371	071364	047117	051124	046117
10372	071372	042514	020122	047514
10373	071400	044507	020103	042524
10374	071406	052123	026440	050040
10375	071414	051101	020124	006461
10376	071422	005012	000	
10377	071425	015	042412	052116
10378	071432	051105	042040	044522
10379	071440	042526	040440	042104
10380	071446	042522	051523	020072
10381	071454	000		
10382	071455	111	053116	046101
10383	071462	042111	040440	042104
10384	071470	042522	051523	005015
10385	071476	000		
10386	071477	015	050012	051117
10387	071504	020124	020101	042101
10388	071512	051104	051505	020123
10389	071520	051511	020072	000
10390	071525	015	050012	051117
10391	071532	020124	020102	042101
10392	071540	051104	051505	020123
10393	071546	051511	020072	000
10394	071553	015	051412	051531
10395	071560	042524	020115	052515
10396	071566	052123	044040	053101
10397	071574	020105	046047	020047
10398	071602	051117	023440	023520
10399	071610	041440	047514	045503
10400	071616	005015	000012	
10401	071622	042412	052116	051105
10402	071630	052040	051505	020124
10403	071636	035043	000040	
10404	071642	047111	040526	044514
10405	071650	020104	042524	052123
10406	071656	047040	046525	042502
10407	071664	006522	000012	
10408	071670	042440	051122	051117
10409	071676	006523	000012	
10410	071702	005015	052012	042510
10411	071710	050040	042522	042523
10412	071716	052116	040440	042104
10413	071724	042522	051523	047440

;;*****

.SBTTL TELETYPE MESSAGES

;;*****

TITLE: .ASCII <CR><LF>'<LF>/MAINDEC-11-DZRJE-A/<CR><LF>

.ASCIZ RMO3 DUAL CONTROLLER LOGIC TEST - PART 1<CR><LF><LF>

ENTERA: .ASCIZ <CR><LF>/ENTER DRIVE ADDRESS: /

ADRERR: .ASCIZ /INVALID ADDRESS/<CR><LF>

PORTAIS: .ASCIZ <CR><LF>/PORT A ADDRESS IS: /

PORTBIS: .ASCIZ <CR><LF>/PORT B ADDRESS IS: /

NOCLOCK: .ASCIZ <CR><LF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CR><LF><LF>

TESTNO: .ASCIZ <LF>/ENTER TEST #: /

BADNO: .ASCIZ /INVALID TEST NUMBER/<CR><LF>

TSTERR: .ASCIZ / ERRORS/<CR><LF>

ADDRIS: .ASCIZ <CR><LF><LF>/THE PRESENT ADDRESS OF THE RH11 (RMCS1) IS: /

10414	071732	020106	044124	020105
10415	071740	044122	030461	024040
10416	071746	046522	051503	024461
10417	071754	044440	035123	000040
10418	071762	042412	052116	051105
10419	071770	047040	053505	051040
10420	071776	030510	020061	042101
10421	072004	051104	051505	035123
10422	072012	000040		
10423				
10424				
10425				
10426				
10427				
10428				
10429				
10430	072014	051127	047117	020107
10431	072022	051104	053111	020105
10432	072030	054524	042520	000
10433				
10434	072035	104	044522	042526
10435	072042	047040	052117	047440
10436	072050	020116	044514	042516
10437	072056	000		
10438				
10439	072057	123	051105	040511
10440	072064	020114	052516	041115
10441	072072	051105	051040	040505
10442	072100	020104	044124	047522
10443	072106	043525	020110	040505
10444	072114	044103	050040	051117
10445	072122	020124	047516	020124
10446	072130	044124	020105	040523
10447	072136	042515	000	
10448				
10449	072141	104	044522	042526
10450	072146	047040	052117	051440
10451	072154	044505	042532	020104
10452	072162	054502	050040	051117
10453	072170	000124		
10454				
10455	072172	051127	047117	020107
10456	072200	052123	052101	051525
10457	072206	051440	042505	020116
10458	072214	054502	052040	042510
10459	072222	051440	044505	044532
10460	072230	043516	050040	051117
10461	072236	000124		
10462				

NTRH11: .ASCIZ <LF>/ENTER NEW RH11 ADDRESS: /

;;*****

.SBTTL TEST ERROR MESSAGES

;;*****

EM1: .ASCIZ /WRONG DRIVE TYPE/

EM2: .ASCIZ /DRIVE NOT ON LINE/

EM3: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/

EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/

EM5: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/

10463	072240	042522	044507	052123	EM6: .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
10464	072246	051105	041440	047117	
10465	072254	042524	052116	020123	
10466	072262	042527	042522	051440	
10467	072270	042505	020116	054502	
10468	072276	047440	050120	051517	
10469	072304	052111	020105	047520	
10470	072312	052122	026440	042040	
10471	072320	044522	042526	053440	
10472	072326	051501	051440	044505	
10473	072334	042532	000104		
10474					
10475	072340	042522	044507	052123	EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
10476	072346	051105	041440	047117	
10477	072354	042524	052116	020123	
10478	072362	051127	047117	020107	
10479	072370	043101	042524	020122	
10480	072376	042522	042514	051501	
10481	072404	020105	051117	052040	
10482	072412	046511	047505	052125	
10483	072420	000			
10484					
10485	072421	122	043505	051511	EM10: .ASCIZ /REGISTER CONTENTS WRONG/
10486	072426	042524	020122	047503	
10487	072434	052116	047105	051524	
10488	072442	053440	047522	043516	
10489	072450	000			
10490					
10491	072451	103	047117	051124	EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
10492	072456	046117	041040	051525	
10493	072464	050040	051101	052111	
10494	072472	020131	051105	047522	
10495	072500	020122	042522	042101	
10496	072506	047111	020107	047111	
10497	072514	044504	040503	042524	
10498	072522	020104	042522	044507	
10499	072530	052123	051105	000	
10500					
10501	072535	104	044522	042526	EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
10502	072542	047040	052117	051440	
10503	072550	044505	042532	020104	
10504	072556	054502	042040	044522	
10505	072564	042526	041440	042514	
10506	072572	051101	041440	046517	
10507	072600	040515	042116	000	
10508					
10509	072605	122	040505	044504	EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
10510	072612	020116	051120	051505	
10511	072620	052105	042040	042517	
10512	072626	020123	047516	020124	
10513	072634	042523	020124	047526	
10514	072642	052514	042515	053040	
10515	072650	046101	042111	043040	
10516	072656	051117	052040	042510	
10517	072664	050040	051117	000124	
10518					

L15

MD-11-DZRMG-A,RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 193
TEST ERROR MESSAGES

SEQ 0195

10519	072672	047526	052514	042515	EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
10520	072700	053040	046101	042111	
10521	072706	051440	052105	047440	
10522	072714	020116	044124	020105	
10523	072722	051127	047117	020107	
10524	072730	047520	052122	000	
10525					
10526	072735	101	052124	020116	EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
10527	072742	044502	020124	051127	
10528	072750	047117	020107	043101	
10529	072756	042524	020122	044524	
10530	072764	042515	052517	020124	
10531	072772	020055	042522	052521	
10532	073000	051505	020124	047516	
10533	073006	020124	042523	000124	
10534					
10535	073014	052101	047124	041040	EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
10536	073022	052111	053440	047322	
10537	073030	043516	040440	052106	
10538	073036	051105	051040	046105	
10539	073044	040505	042523	026440	
10540	073052	051040	050505	042525	
10541	073060	052123	051440	052105	
10542	073066	000			
10543					
10544	073067	101	052124	020116	EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
10545	073074	044502	020124	051127	
10546	073102	047117	020107	043101	
10547	073110	042524	020122	042522	
10548	073116	042514	051501	020105	
10549	073124	020055	042522	052521	
10550	073132	051505	020124	047516	
10551	073140	020124	042523	000124	
10552					
10553	073146	051104	053111	020105	EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
10554	073154	047516	020124	042523	
10555	073162	055111	042105	053440	
10556	073170	042510	020116	052101	
10557	073176	047124	041040	052111	
10558	073204	043040	051117	050040	
10559	073212	051117	020124	046103	
10560	073220	040505	042522	000104	

M15

MO-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 194
TEST ERROR MESSAGES

SEQ 0196

10561					
10562	073226	051104	053111	020105	EM21: .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
10563	073204	042523	055111	042105	
10564	073242	053440	042510	020116	
10565	073250	042532	047522	053440	
10566	073256	044522	052124	047105	
10567	073264	044440	020116	052101	
10568	073272	047124	041040	052111	
10569	073300	000			
10570					
10571	073301	104	044522	042526	EM22: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
10572	073306	047040	052117	044440	
10573	073314	020116	042516	052125	
10574	073322	040522	020114	043101	
10575	073330	042524	020122	044524	
10576	073336	042515	052517	020124	
10577	073344	020055	042522	052521	
10578	073352	051505	020124	047516	
10579	073360	020124	042523	000124	
10580					
10581	073366	044524	042515	052517	EM23: .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
10582	073374	020124	046103	040505	
10583	073402	042522	020104	044124	
10584	073410	020105	051104	053111	
10585	073416	023505	020123	051105	
10586	073424	047522	020122	044502	
10587	073432	000124			
10588					
10589	073434	042522	042514	051501	EM24: .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
10590	073442	020105	047503	046515	
10591	073450	047101	020104	042522	
10592	073456	042514	051501	042105	
10593	073464	042040	044522	042526	
10594	073472	053440	052111	020110	
10595	073500	051105	047522	051522	
10596	073506	051440	052105	000	
10597					
10598	073513	124	046511	047505	EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
10599	073520	052125	047440	042516	
10600	073526	051455	047510	020124	
10601	073534	044504	020104	047516	
10602	073542	020124	042522	051124	
10603	073550	043511	042507	000122	
10604					
10605	073556	051104	053111	020105	EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
10606	073564	047516	020124	047111	
10607	073572	047040	052505	051124	
10608	073600	046101	040440	052106	
10609	073606	051105	051040	046105	
10610	073614	040505	042523	026440	
10611	073622	051040	050505	042525	
10612	073630	052123	047040	052117	
10613	073636	051440	052105	000	
10614					
10615	073643	122	043505	051511	EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
10616	073650	042524	020122	051127	

10617	073656	047117	020107	043101
10618	073664	042524	020122	042522
10619	073672	042514	051501	020105
10620	073700	044527	044124	051040
10621	073706	050505	042525	052123
10622	073714	051440	052105	000
10623				
10624	073721	104	044522	042526
10625	073726	051440	044505	042532
10626	073734	020104	054502	051040
10627	073742	046105	040505	042523
10628	073750	041440	046517	040515
10629	073756	042116	044440	051523
10630	073764	042525	020104	044127
10631	073772	047105	042040	044522
10632	074000	042526	044440	020116
10633	074006	042516	052125	040522
10634	074014	000114		
10635				
10636	074016	051104	053111	020105
10637	074024	047111	047040	052505
10638	074032	051124	046101	040440
10639	074040	052106	051105	051040
10640	074046	046105	040	042523
10641	074054	026440	051040	050505
10642	074062	042525	052123	051440
10643	074070	052105	000	
10644				
10645	074073	101	052124	020116
10646	074100	044502	020124	051127
10647	074106	047117	020107	043101
10648	074114	042524	020122	042522
10649	074122	040503	044514	051102
10650	074130	052101	020105	047503
10651	074136	046515	047101	000104
10652				
10653	074144	051104	053111	020105
10654	074152	042522	052524	047122
10655	074160	042105	052040	020117
10656	074166	042516	052125	040522
10657	074174	020114	043111	042040
10658	074202	044522	042526	041440
10659	074210	042514	051101	043440
10660	074216	053111	047105	053440
10661	074224	044510	042514	042040
10662	074232	044522	042526	051440
10663	074240	044505	042532	000104
10664				
10665	074246	051104	053111	020105
10666	074254	042522	052524	047122
10667	074262	042105	052040	020117
10668	074270	042516	052125	040522
10669	074276	020114	043111	046440
10670	074304	051501	041123	051525
10671	074312	044440	044516	020124
10672	074320	044507	042526	020116

EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/

EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/

EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/

EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/

EM34: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/

10673	074326	044127	046111	020105	
10674	074334	051104	053111	020105	
10675	074342	042523	055111	042105	
10676	074350	000			
10677					
10678	074351	124	046511	047505	EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
10679	074356	052125	047440	042516	
10680	074364	051440	047510	020124	
10681	074372	044506	042522	020104	
10682	074400	044527	044124	052517	
10683	074406	020124	042522	044507	
10684	074414	052123	051105	040440	
10685	074422	041503	051515	000123	
10686					
10687	074430	044524	042515	052517	EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
10688	074436	020124	040510	020123	
10689	074444	047516	020124	041517	
10690	074452	052503	051122	042105	
10691	074460	053440	052111	044510	
10692	074466	020116	020062	042523	
10693	074474	047503	042116	000123	
10694					
10695	074502	051104	053111	020105	EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
10696	074510	051511	047040	047117	
10697	074516	042455	044530	052123	
10698	074524	047105	020124	023450	
10699	074532	042516	023504	041040	
10700	074540	052111	051440	052105	
10701	074546	000051			
10702					
10703	074550	052101	047124	041040	EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
10704	074556	052111	043040	051117	
10705	074564	050040	051117	020124	
10706	074572	047516	020124	042522	
10707	074600	042523	020124	054502	
10708	074606	046440	051501	041123	
10709	074614	051525	041440	042514	
10710	074622	051101	000		
10711					
10712	074625	124	046511	047505	EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
10713	074632	052125	041440	042514	
10714	074640	051101	042105	052040	
10715	074646	042510	040440	052124	
10716	074654	047105	044524	047117	
10717	074662	041040	052111	000	
10718					
10719	074667	104	044522	042526	EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
10720	074674	047040	052117	044440	
10721	074702	020116	042516	052125	
10722	074710	040522	020114	051117	
10723	074716	051440	044505	042532	
10724	074724	020104	043101	042524	
10725	074732	020122	052101	047124	
10726	074740	041040	052111	053440	
10727	074746	044522	052124	047105	
10728	074754	000			

TEST ERROR MESSAGES

10729					
10730	074755	104	044522	042526	EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
10731	074762	044440	020116	042516	
10732	074770	052125	040522	020114	
10733	074776	043101	042524	020122	
10734	075004	052101	042524	052116	
10735	075012	047511	020116	044502	
10736	075020	020124	051127	052111	
10737	075026	042524	000116		
10738					
10739	075032	051127	052111	020105	EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
10740	075040	052101	042524	052116	
10741	075046	047511	020116	044502	
10742	075054	020124	044504	020104	
10743	075062	047516	020124	042523	
10744	075070	020124	047520	052122	
10745	075076	051040	050505	042525	
10746	075104	052123	000		
10747					
10748	075107	103	047117	051124	EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
10749	075114	046117	042514	020122	
10750	075122	042523	042514	052103	
10751	075130	051440	044527	041524	
10752	075136	020110	047117	042040	
10753	075144	044522	042526	047040	
10754	075152	052117	044440	020116	
10755	075160	040447	041057	000047	
10756					
10757	075166	040503	023516	020124	EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
10758	075174	041501	042503	051523	
10759	075202	042040	044522	042526	
10760	075210	052040	051110	052517	
10761	075216	044107	042440	052111	
10762	075224	042510	020122	047520	
10763	075232	052122	000		
10764					
10765	075235	101	052124	020116	EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
10766	075242	044502	020124	047506	
10767	075250	020122	042523	055111	
10768	075256	047111	020107	047520	
10769	075264	052122	047040	052117	
10770	075272	041440	042514	051101	
10771	075300	042105	041040	020131	
10772	075306	040515	051523	052502	
10773	075314	020123	047111	052111	
10774	075322	000			
10775					
10776	075323	101	052124	020116	EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR/
10777	075330	044502	020124	047506	
10778	075336	020122	050117	047520	
10779	075344	044523	042524	050040	
10780	075352	051117	020124	046103	
10781	075360	040505	042522	020104	
10782	075366	054502	042040	044522	
10783	075374	042526	041440	042514	
10784	075402	051101	000		

10785					
10786	075405	101	052124	020116	EMS1: .ASCIZ /ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
10787	075412	044502	020124	047516	
10788	075420	020124	046103	040505	
10789	075426	042522	020104	054502	
10790	075434	046440	051501	041123	
10791	075442	051525	044440	044516	
10792	075450	026124	042040	044522	
10793	075456	042526	044440	020116	
10794	075464	042516	052125	040522	
10795	075472	000114			
10796					
10797	075474	044124	020105	052101	EMS2: .ASCIZ /THE ATTN BIT SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
10798	075502	047124	041040	052111	
10799	075510	051440	052105	040440	
10800	075516	052106	051105	052040	
10801	075524	046511	047505	052125	
10802	075532	053440	052111	020110	
10803	075540	047516	051040	050505	
10804	075546	042525	052123	023040	
10805	075554	023440	051105	023522	
10806	075562	051440	052105	000	
10807					
10808	075567	103	047101	052047	EMS3: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
10809	075574	051040	040505	020104	
10810	075602	044124	020105	052101	
10811	075610	047124	041040	052111	
10812	075616	043040	047522	020115	
10813	075624	044124	020105	047447	
10814	075632	050120	051517	052111	
10815	075640	023505	050040	051117	
10816	075646	000124			
10817					
10818	075650	042522	042514	051501	EMS4: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
10819	075656	020105	047503	046515	
10820	075664	047101	020104	042522	
10821	075672	047503	047107	055111	
10822	075700	042105	053440	042510	
10823	075706	020116	051511	052523	
10824	075714	042105	041040	020131	
10825	075722	047516	026516	042523	
10826	075730	055111	047111	020107	
10827	075736	047520	052122	000	
10828					
10829	075743	124	046511	047505	EMS5: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
10830	075750	052125	047440	042516	
10831	075756	051455	047510	020124	
10832	075764	051511	046040	051505	
10833	075772	020123	044124	047101	
10834	076001	032440	030060	046440	
10835	076006	000123			
10836					
10837	076010	044122	030461	042040	EMS6: .ASCIZ /R411 DIDN'T RESPOND TO ADDRESSING/
10838	076016	042111	023516	020124	
10839	076024	042522	050123	047117	
10840	076032	020104	047524	040440	

10841	076040	042104	042522	051523	
10842	076046	047111	000107		
10843					
10844	076052	047520	052122	051040	EM57: .ASCIZ /PORT REQUEST FLOP(S) WRONG STATE/
10845	076060	050505	042525	052123	
10846	076066	043040	047514	024120	
10847	076074	024523	053440	047522	
10848	076102	043516	051440	040524	
10849	076110	042524	000		
10850					
10851	076113	101	052124	047105	EM60: .ASCIZ /ATTENTION NOT RESET BY WRITING RMAS/
10852	076120	044524	047117	047040	
10853	076126	052117	051040	051505	
10854	076134	052105	041040	020131	
10855	076142	051127	052111	047111	
10856	076150	020107	046522	051501	
10857	076156	000			
10858					
10859	076157	101	052124	047105	EM61: .ASCIZ /ATTENTION NOT RESET BY GO/
10860	076164	044524	047117	047040	
10861	076172	052117	051040	051505	
10862	076200	052105	041040	020131	
10863	076206	047507	000		
10864					
10865	076211	101	052124	047105	EM62: .ASCIZ /ATTENTION RESET BY GO WHEN NOT SEIZED/
10866	076216	044524	047117	051040	
10867	076224	051505	052105	041040	
10868	076232	020131	047507	053440	
10869	076240	042510	020116	047516	
10870	076246	020124	042523	055111	
10871	076254	042105	000		
10872					
10873	076257	104	044522	042526	EM63: .ASCIZ /DRIVE SEIZED BY UNIT READY CHANGE/
10874	076264	051440	044505	042532	
10875	076272	020104	054502	052440	
10876	076300	044516	020124	042522	
10877	076306	042101	020131	044103	
10878	076314	047101	042507	000	
10879					
10880	076321	101	052124	047105	EM64: .ASCIZ /ATTENTION NOT SET BY UNIT READY CHANGE/
10881	076326	044524	047117	047040	
10882	076334	052117	051440	052105	
10883	076342	041040	020131	047125	
10884	076350	052111	051040	040505	
10885	076356	054504	041440	040510	
10886	076364	043516	000105		
10887					
10888	076370	047526	052514	042515	EM65: .ASCIZ /VOLUME VALID NOT RESET BY UNIT READY/
10889	076376	053040	046101	042111	
10890	076404	047040	052117	051040	
10891	076412	051505	052105	041040	
10892	076420	020131	047125	052111	
10893	076426	051040	040505	054504	
10894	076434	000			
10895					
10896					

10953	077104	020043	042440	051122
10954	077112	050040	020103	050040
10955	077120	051117	020124	020043
10956	077126	051040	043505	040440
10957	077134	051104	041440	047117
10958	077142	042524	052116	000123
10959	077150	020040	020040	020040
10960	077156	020040	020040	020040
10961	077164	020040	020040	042523
10962	077172	055111	020105	020040
10963	077200	051105	047522	006522
10964	077206	012		
10965	077207	124	051505	020124
10966	077214	020043	042440	051122
10967	077222	050040	020103	050040
10968	077230	051117	020124	020043
10969	077236	050040	051117	020124
10970	077244	020043	051040	043505
10971	077252	040440	051104	041440
10972	077260	047117	042524	052116
10973	077266	000123		
10974	077270	020040	020040	020040
10975	077276	020040	020040	020040
10976	077304	020040	020040	042522
10977	077312	051514	043516	020040
10978	077320	042523	055111	006505
10979	077326	012		
10980	077327	124	051505	020124
10981	077334	020043	042440	051122
10982	077342	050040	020103	050040
10983	077350	051117	020124	020043
10984	077356	050040	051117	020124
10985	077364	000043		
10986	077366	020040	020040	020040
10987	077374	020040	020040	020040
10988	077402	020040	020040	042523
10989	077410	055111	006505	012
10990	077415	124	051505	020124
10991	077422	020043	042440	051122
10992	077430	050040	020103	050040
10993	077436	051117	020124	020043
10994	077444	051040	043505	040440
10995	077452	051104	041440	047117
10996	077460	042524	052116	000123
10997	077466	020040	020040	020040
10998	077474	020040	020040	020040
10999	077502	020040	020040	042522
11000	077510	051514	043516	005015
11001	077516	042524	052123	021440
11002	077524	020040	051105	020122
11003	077532	041520	020040	047520
11004	077540	052122	021440	000
11005	077545	040	020040	020040
11006	077552	020040	020040	020040
11007	077560	020040	020040	051040
11008	077566	046105	047123	020107

DH13: .ASCII / SEIZE ERROR<<CR><LF>

.ASCIZ /TEST # ERR PC PORT # PORT # REG ADR CONTENTS/

DH22: .ASCII / RELSNG SEIZE<<CR><LF>

.ASCIZ /TEST # ERR PC PORT # PORT #/

DH23: .ASCII / SEIZE<<CR><LF>

.ASCIZ /TEST # ERR PC PORT # REG ADR CONTENTS/

DH26: .ASCII / RELSNG<<CR><LF>

.ASCIZ /TEST # ERR PC PORT #/

DH31: .ASCII / RELSNG RQSTNG<<CR><LF>

11009	077574	051040	051521	047124					
11010	077602	006507	012						
11011	077605	124	051505	020124	.ASCIZ	/TEST #	ERR PC	PORT #	PORT #/
11012	077612	020043	042440	051122					
11013	077620	050040	020103	050040					
11014	077626	051117	020124	020043					
11015	077634	050040	051117	020124					
11016	077642	000043							
11017	077644	042524	052123	021440	DH36:	.ASCIZ	/TEST #	ERR PC	PORT #/
11018	077652	020040	051105	020122					
11019	077660	041520	020040	047520					
11020	077666	052122	021440	000					
11021	077673	124	051505	020124	DH42:	.ASCIZ	/TEST #	ERR PC/	
11022	077700	020043	042440	051122					
11023	077706	050040	000103						
11024	077712	020040	020040	020040	DH44:	.ASCII	/	RELSNG	ERROR/<CR><LF>
11025	077720	020040	020040	020040					
11026	077726	020040	020040	042522					
11027	077734	051514	043516	020040					
11028	077742	051105	047522	006522					
11029	077750	012							
11030	077751	124	051505	020124	.ASCIZ	/TEST #	ERR PC	PORT #	PORT #/
11031	077756	020043	042440	051122					
11032	077764	050040	020103	050040					
11033	077772	051117	020124	020043					
11034	00070	050040	051117	020124					
11035	100006	000043							
11036	100010	020040	020040	020040	DH46:	.ASCII	/	PORT A	PORT B/<CR><LF>
11037	100016	020040	020040	020040					
11038	100024	020040	020040	047520					
11039	100032	052122	040440	020040					
11040	100040	047520	052122	041040					
11041	100046	005015							
11042	100050	042524	052123	021440	.ASCIZ	/TEST #	ERR PC	RPDS1	RMDS1/
11043	100056	020040	051105	020122					
11044	100064	041520	020040	050122					
11045	100072	051504	020061	020040					
11046	100100	046522	051504	000061					
11047	100106	042524	052123	021440	DH55:	.ASCIZ	/TEST #	ERR PC	PORT #
11048	100114	020040	051105	020122					TIMEOUT VALUE (IN MS)/
11049	100122	041520	020040	047520					
11050	100130	052122	021440	020040					
11051	100136	044524	042515	052517					
11052	100144	020124	040526	052514					
11053	100152	020105	044450	020116					
11054	100160	051515	000051						
11055	100164	051044	040515	051104	DH56:	.ASCIZ	/\$RMADR/		
11056	100172	000							
11057	100173	124	051505	020124	DH57:	.ASCII	/TEST #	ERR PC	PORT A
11058	100200	020043	042440	051122					PORT B/
11059	100206	050040	020103	020040					
11060	100214	020040	050040	051117					
11061	100222	020124	020101	020040					
11062	100230	020040	020040	020040					
11063	100236	050040	051117	020124					
11064	100244	102							


```

11086 100410 001246 001116 001242 DT7: .WORD TSTNUM, $ERRPC, SEIZPT, PTNBR, $BDADR, $GDDAT, $BDDAT, 0
11087 100416 001240 001122 001124
11088 100424 001126 000000
11089 100430 001246 001116 001242 DT13: .WORD TSTNUM, $ERRPC, SEIZPT, PTNBR, $BDADR, $BDDAT, 0
11090 100436 001240 001122 001126
11091 100444 000000
11092 100446 001246 001116 001242 DT22: .WORD TSTNUM, $ERRPC, SEIZPT, PTNBR, 0
11093 100454 001240 000000
11094 100460 001246 001116 001242 DT23: .WORD TSTNUM, $ERRPC, SEIZPT, $BDADR, $BDDAT, 0
11095 100466 001122 001126 000000
11096 100474 001246 001116 001242 DT31: .WORD TSTNUM, $ERRPC, SEIZPT, OPPRT, 0
11097 100502 001244 000000
11098 100506 001246 001116 001242 DT36: .WORD TSTNUM, $ERRPC, SEIZPT, 0
11099 100514 000000
11100 100516 001246 001116 001240 DT37: .WORD TSTNUM, $ERRPC, PTNBR, 0
11101 100524 000000
11102 100526 001246 001116 000000 DT42: .WORD TSTNUM, $ERRPC, 0
11103 100534 001246 001116 001170 DT46: .WORD TSTNUM, $ERRPC, $TMP2, $TMP3, 0
11104 100542 001172 000000
11105 100546 001246 001116 001244 DT54: .WORD TSTNUM, $ERRPC, OPPRT, SEIZPT, 0
11106 100554 001242 000000
11107 100560 001246 001116 001242 DT55: .WORD TSTNUM, $ERRPC, SEIZPT, TIME, 0
11108 100566 001256 000000
11109 100572 001304 000000 DT56: .WORD $RMADR, 0
11110 100576 001246 001116 001164 DT57: .WORD TSTNUM, $ERRPC, $TMP0, $TMP1, $TMP2, $TMP3, 0
11111 100604 001166 001170 001172
11112 100612 000000
11113
11114 100614 000 000 000 DF1: .BYTE 0,0,0,0,0
11115 100617 000 000
11116 100621 000 000 000 DF5: .BYTE 0,0,0,0,0,0
11117 100624 000 000 000
11118 100627 000 000 000 DF7: .BYTE 0,0,0,0,0,0,0
11119 100632 000 000 000
11120 100635 000
11121 100636 000 000 000 DF31: .BYTE 0,0,0,0
11122 100641 000
11123 100642 000 000 000 DF36: .BYTE 0,0,0
11124 100645 000 000 DF42: .BYTE 0,0
11125 100647 000 000 000 DF55: .BYTE 0,0,0,1
11126 100652 001
11127 100653 000 DF56: .BYTE 0
11128
11129 .EVEN
11130
11131
11132 ;*****
11133 .SBTTL CONSTANTS, TABLES, ETC
11134
11135 ;*****
11136
11137 ;TABLE OF TEST STARTING ADDRESSES
11138
11139
11140 100654 003164 TSTADR: .WORD TST1 ;STARTING ADDRESS OF TEST 1
11141 100656 004572 .WORD TST2 ;STARTING ADDRESS OF TEST 2

```

11142	100660	006140	.WORD	TST3	; STARTING ADDRESS OF TEST 3
11143	100662	007506	.WORD	TST4	; STARTING ADDRESS OF TEST 4
11144	100664	010636	.WORD	TST5	; STARTING ADDRESS OF TEST 5
11145	100666	011766	.WORD	TST6	; STARTING ADDRESS OF TEST 6
11146	100670	012440	.WORD	TST7	; STARTING ADDRESS OF TEST 7
11147	100672	013112	.WORD	TST10	; STARTING ADDRESS OF TEST 10
11148	100674	014356	.WORD	TST11	; STARTING ADDRESS OF TEST 11
11149	100676	015622	.WORD	TST12	; STARTING ADDRESS OF TEST 12
11150	100700	016742	.WORD	TST13	; STARTING ADDRESS OF TEST 13
11151	100702	020062	.WORD	TST14	; STARTING ADDRESS OF TEST 14
11152	100704	021462	.WORD	TST15	; STARTING ADDRESS OF TEST 15
11153	100706	023062	.WORD	TST16	; STARTING ADDRESS OF TEST 16
11154	100710	024006	.WORD	TST17	; STARTING ADDRESS OF TEST 17
11155	100712	024732	.WORD	TST20	; STARTING ADDRESS OF TEST 20
11156	100714	026000	.WORD	TST21	; STARTING ADDRESS OF TEST 21
11157	100716	027046	.WORD	TST22	; STARTING ADDRESS OF TEST 22
11158	100720	031122	.WORD	TST23	; STARTING ADDRESS OF TEST 23
11159	100722	031646	.WORD	TST24	; STARTING ADDRESS OF TEST 24
11160	100724	033042	.WORD	TST25	; STARTING ADDRESS OF TEST 25
11161	100726	034236	.WORD	TST26	; STARTING ADDRESS OF TEST 26
11162	100730	035732	.WORD	TST27	; STARTING ADDRESS OF TEST 27
11163	100732	037426	.WORD	TST30	; STARTING ADDRESS OF TEST 30
11164	100734	041122	.WORD	TST31	; STARTING ADDRESS OF TEST 31
11165	100736	042616	.WORD	TST32	; STARTING ADDRESS OF TEST 32
11166	100740	044074	.WORD	TST33	; STARTING ADDRESS OF TEST 33
11167	100742	045220	.WORD	TST34	; STARTING ADDRESS OF TEST 34
11168	100744	046116	.WORD	TST35	; STARTING ADDRESS OF TEST 35
11169	100746	047014	.WORD	TST36	; STARTING ADDRESS OF TEST 36
11170	100750	050012	.WORD	TST37	; STARTING ADDRESS OF TEST 37
11171	100752	051010	.WORD	TST40	; STARTING ADDRESS OF TEST 40
11172	100754	052774	.WORD	TST41	; STARTING ADDRESS OF TEST 41
11173	100756	054760	.WORD	TST42	; STARTING ADDRESS OF TEST 42
11174	100760	056234	.WORD	TST43	; STARTING ADDRESS OF TEST 43
11175	100762	057510	.WORD	TST44	; STARTING ADDRESS OF TEST 44
11176	100764	060340	.WORD	TST45	; STARTING ADDRESS OF TEST 45
11177	100766	061170	.WORD	TST46	; STARTING ADDRESS OF TEST 46

;ATTENTION BIT TABLE

11181	100770	001	ATABIT: .BYTE	1	; ATTENTION BIT FOR DRIVE 0
11182	100771	002	.BYTE	2	; ATTENTION BIT FOR DRIVE 1
11183	100772	004	.BYTE	4	; ATTENTION BIT FOR DRIVE 2
11184	100773	010	.BYTE	10	; ATTENTION BIT FOR DRIVE 3
11185	100774	020	.BYTE	20	; ATTENTION BIT FOR DRIVE 4
11186	100775	040	.BYTE	40	; ATTENTION BIT FOR DRIVE 5
11187	100776	100	.BYTE	100	; ATTENTION BIT FOR DRIVE 6
11188	100777	200	.BYTE	200	; ATTENTION BIT FOR DRIVE 7
11189					
11190	101000	'00052	MAXTN: .WORD	\$TN-1	; MAXIMUM TEST NUMBER
11191					
11192		000001	.END		

B01

MD-1, -DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 209
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0210

DH36	077644	1654	1696	1703	1732	1754	1775	1782	1789	1796	1803	1939	11017#	
DH4	076555	1612	10914#											
DH42	077673	1824	1831	11021#										
DH44	077712	1838	11024#											
DH46	100010	1852	11036#											
DH5	076700	1619	1640	1768	1873	1921	1951	10929#						
DH55	100106	1901	11047#											
DH56	100164	1908	11055#											
DH57	100173	1915	11057#											
DH7	076754	1633	1675	1682	1689	1747	1945	10937#						
DISPLA	001142	1507#	1991#	1999#	9661#	9683#								
DISPRE	000174	1456#	1999											
DLT	= 100000	1290#												
DMD	= 000001	1351#	7619	7621	7886	7888								
DPE	= 000010	1396#												
DPR	= 000400	1321#	2257	2289	2380	2440	2608	2668	2845	3012	3146	3230	3342	3394
		3425	3536	3588	3619	3793	3966	4067	4108	4151	4174	4274	4315	4358
		4381	4486	4509	4613	4636	4745	4791	4899	4945	5071	5094	5125	5181
		5204	5235	5334	5452	5519	5634	5701	5862	5909	6090	6137	6322	6370
		6555	6603	7091	7207	7340	7490	7681	7707	7798	7948	7974	8065	8666
		8745	8903	9049	9177	9208	9373	9404						
DRQ	= 004200	1381#												
DRY	= 000200	1320#	2257	2289	2380	2440	2608	2668	2845	3012	3146	3230	3342	3394
		3425	3536	3588	3619	3793	3966	4067	4108	4151	4174	4274	4315	4358
		4381	4486	4509	4613	4636	4745	4791	4899	4945	5071	5094	5125	5181
		5204	5235	5334	5452	5519	5634	5701	5862	5909	6090	6137	6322	6370
		6555	6603	7091	7207	7307	7340	7457	7490	7681	7707	7798	7948	7974
		8065	8606	8745	8903	9049	9177	9208	9373	9404				
DSWR	= 177570	1155#	1506	1990										
DTE	= 010000	1344#												
DT00	= 000001	1372#												
DT01	= 000002	1373#												
DT02	= 000004	1374#												
DT03	= 000010	1375#												
DT04	= 000020	1376#												
DT05	= 000040	1377#												
DT06	= 000100	1378#												
DT07	= 000200	1379#												
DT08	= 000400	1380#												
DT1	100326	1592	1599	1648	1846	1888	11077#							
DT13	100430	1627	1662	1669	1867	1881	1934	11089#						
DT22	100446	1711	1741	11092#										
DT23	100460	1718	1725	1811	1818	1860	1928	11094#						
DT3	100342	1606	11079#											
DT31	100474	1762	1839	11096#										
DT36	100506	1733	1755	1776	1783	1797	1940	11098#						
DT37	100516	1655	1697	1704	1790	1804	11100#							
DT42	100526	1825	1832	11102#										
DT46	100534	1853	11103#											
DT5	100356	1620	1641	1769	1874	1922	1952	11081#						
DT54	100546	1895	11105#											
DT55	100560	1902	11107#											
DT56	100572	1909	11109#											
DT57	100576	1916	11110#											
DT6	100374	11084#												
DT7	100410	1613	1634	1676	1683	1690	1748	1946	11086#					

DVA	= 004000	1310#		
DVC	= 000200	1397#		
ECH	= 000100	1338#		
ECI	= 004000	1408#		
EMTVEC	= 000030	1244#	1976*	1977*
EM1	072014	1590	10430#	
EM10	072421	1639	10485#	
EM11	072451	1646	10491#	
EM12	072535	1653	10501#	
EM13	072605	1660	10509#	
EM14	072672	1667	10519#	
EM15	072735	1674	10526#	
EM16	073014	1681	10535#	
EM17	073067	1688	10544#	
EM2	072035	1597	10434#	
EM20	073146	1695	10553#	
EM21	073226	1702	10562#	
EM22	073301	1709	10571#	
EM23	073366	1716	10581#	
EM24	073434	1723	10589#	
EM25	073513	1731	10598#	
EM26	073556	1739	10605#	
EM27	073643	1746	10615#	
EM3	072057	1604	10439#	
EM30	073721	1753	10624#	
EM31	074016	1760	10636#	
EM32	074073	1767	10645#	
EM33	074144	1774	10653#	
EM34	074246	1781	10665#	
EM35	074351	1788	10678#	
EM36	074430	1795	10687#	
EM37	074502	1802	10695#	
EM4	072141	1611	10449#	
EM40	074550	1809	10703#	
EM41	074625	1816	10712#	
EM42	074667	1823	10719#	
EM43	074755	1830	10730#	
EM44	075032	1837	10739#	
EM45	075107	1844	10748#	
EM46	075166	1851	10757#	
EM47	075235	1858	10765#	
EM5	072172	1618	10455#	
EM50	075323	1865	10776#	
EM51	075405	1872	10786#	
EM52	075474	1879	10797#	
EM53	075567	1886	10808#	
EM54	075650	1893	10818#	
EM55	075743	1900	10829#	
EM56	076010	1907	10837#	
EM57	076052	1914	10844#	
EM6	072240	1625	10463#	
EM60	076113	1920	10851#	
EM61	076157	1926	10859#	
EM62	076211	1932	10865#	
EM63	076257	1938	10873#	
EM64	076321	1944	10880#	

E01

MD-11-DZRMG-A, RM03 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 212
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0213

NED = 010000	1287#	2161	2165	2182	2186									
NEM = 004000	1286#													
NOATA = 000001	1956#	2471	2477	2699	2705	2876	2882	3043	3049	3179	3185	3263	3269	
	3458	3464	3652	3658	3826	3832	3999	4005	4207	4213	4414	4420	4542	
	4548	4669	4675	4824	4830	4978	4984	5158	5164	5268	5274	5367	5373	
	5439#	5506#	5552	5557	5621#	5688#	5734	5739	5745#	5942	5948	6170	6176	
	6403	6409	6636	6642	6757#	6798	6803	6854#	6980	6985	7078#	7124	7130	
	7135#	7194#	7240	7246	7251#	7373	7378	7523	7528	7692#	7738	7744	7829	
	7835	7959#	8005	8011	8096	8102	8109#	8235#	8274	8280	8285#	8439#	8478	
	8484	8489#	8639	8644	8778	8783	8897#	8936	8942	8947#	9043#	9082	9088	
	9093#	9241	9246	9437	9442									
NOCLOC 071553	2053	10394#												
NOSEIZ 001252	1549#													
NTRH11 071762	2098	10418#												
OFD = 000200	1406#													
OM = 000001	1318#	2379	2443	2448	2607	2671	2676	2848	2853	3015	3020	3149	3154	
	3233	3238	3341	3428	3433	3535	3622	3627	3796	3801	3969	3974	4066	
	4177	4182	4273	4384	4389	4485	4512	4517	4612	4639	4644	4744	4794	
	4799	4898	4948	4953	5128	5133	5238	5243	5337	5342	5522	5527	5704	
	5709	5861	5912	5917	6089	6140	6145	6321	6373	6378	6554	6606	6611	
	6768	6773	6950	6955	7094	7099	7210	7215	7343	7348	7493	7498	7710	
	7715	7801	7806	7977	7982	8068	8073	8244	8249	8448	8453	8609	8614	
	8748	8753	8906	8911	9052	9057	9211	9216	9407	9412				
OPE = 020000	1401#													
OPI = 020000	1345#													
OPPRT 001244	1546#	2366#	2594#	2813#	2980#	3328#	3522#	3715#	3888#	4053#	4260#	4472#	4599#	
	4731#	4885#	5083#	5193#	5430#	5612#	5848#	6076#	6308#	6541#	7620#	7887#	8154#	
	8358#	8566#	8705#	8851#	8997#	9149#	9345#	11084	11096	11105				
OR = 000200	1282#													
PAR = 000010	1335#													
PAT = 000020	1279#													
PGE = 002000	1285#													
PGM = 001000	1322#	2242	2244	2248	2274	2276	2280	2380	2440	2608	2668	2845	3012	
	3146	3230	3342	3394	3425	3536	3588	3619	3793	3966	4067	4108	4151	
	4174	4274	4315	4358	4381	4486	4509	4613	4636	4745	4791	4899	4945	
	5071	5094	5125	5181	5204	5235	5334	5452	5519	5634	5701	5862	5909	
	6090	6137	6322	6370	6555	6603	7091	7207	7340	7490	7681	7707	7798	
	7948	7974	8065	8606	8745	8903	9049	9177	9208	9373	9404			
PIP = 020000	1326#	2379	2443	2448	2607	2671	2676	2848	2853	3015	3020	3149	3154	
	3233	3238	3341	3428	3433	3535	3622	3627	3796	3801	3969	3974	4066	
	4177	4182	4273	4384	4389	4485	4512	4517	4612	4639	4644	4744	4794	
	4799	4898	4948	4953	5128	5133	5238	5243	5337	5342	5522	5527	5704	
	5709	5861	5912	5917	6089	6140	6145	6321	6373	6378	6554	6606	6611	
	6768	6773	6950	6955	7094	7099	7210	7215	7343	7348	7493	7498	7710	
	7715	7801	7806	7977	7982	8068	8073	8244	8249	8448	8453	8609	8614	
	8748	8753	8906	8911	9052	9057	9211	9216	9407	9412				
PIRQ = 177772	1154#													
PIRQVE = 000240	1248#													
PORTA 001224	1538#	2018#	2019	2023	2026	2030	2042	2045	2152	2153	2198	2199	2236	
	2237	2304	2361	2362	2376	2377	2441	2462	2464	2470	2592	2593	2594	
	2621	2622	2669	2690	2692	2698	2715	2810	2811	2836	2837	2846	2867	
	2869	2875	2890	2891	2980	3013	3034	3036	3042	3074	3075	3098	3099	
	3128	3129	3130	3147	3168	3170	3178	3231	3252	3254	3262	3310	3326	
	3327	3328	3355	3356	3397	3398	3408	3416	3417	3426	3447	3449	3457	
	3504	3517	3518	3532	3533	3581	3582	3594	3595	3620	3641	3643	3651	
	3700	3712	3713	3741	3742	3762	3763	3784	3785	3794	3815	3817	3825	

FOI

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 213
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0214

3873	3888	3923	3924	3967	3988	3990	3998	4048	4049	4063	4064	4102
4103	4145	4146	4165	4166	4175	4196	4198	4206	4258	4259	4260	4292
4293	4335	4336	4382	4403	4405	4413	4454	4467	4468	4482	4483	4500
4501	4510	4531	4533	4541	4581	4597	4598	4599	4637	4658	4660	4668
4713	4729	4730	4731	4754	4755	4792	4813	4815	4823	4867	4880	4881
4895	4896	4936	4937	4946	4967	4969	4977	5021	5034	5040	5041	5083
5097	5098	5108	5116	5117	5126	5147	5149	5157	5175	5176	5192	5197
5198	5210	5211	5236	5257	5259	5267	308	5335	5356	5358	5366	5414
5430	5431	5432	5455	5456	5466	5488	5489	5510	5511	5520	5541	5543
5551	5596	5609	5610	5627	5628	5640	5641	5653	5654	5702	5723	5725
5733	5781	5791	5798	5799	5843	5844	5858	5859	5900	5901	5910	5931
5933	5941	6009	6019	6046	6047	6074	6075	6076	6138	6159	6161	6169
6185	6186	6241	6251	6258	6259	6303	6304	6318	6319	6361	6362	6371
6392	6394	6402	6474	6484	6511	6512	6539	6540	6541	6604	6625	6627
6635	6652	6653	6700	6710	6717	6718	6766	6787	6789	6797	6818	6819
6885	6895	6900	6901	6948	6969	6971	6972	7010	7011	7048	7056	7092
7113	7115	7123	7164	7175	7184	7185	7186	7198	7199	7208	7229	7231
7239	7287	7295	7296	7297	7331	7332	7341	7362	7364	7372	7437	7491
7512	7514	7522	7532	7533	7617	7618	7698	7699	7708	7729	7731	7737
7799	7820	7822	7828	7887	7975	7996	7998	8004	8016	8017	8056	8057
8066	8087	8089	8095	8139	8151	8152	8178	8179	8242	8263	8265	8273
8309	8343	8358	8370	8371	8446	8467	8469	8477	8493	8494	8550	8563
8564	8607	8628	8630	8638	8689	8705	8729	8730	8746	8767	8769	8777
8835	8851	8852	8953	8904	8925	8927	8935	8981	8994	8995	9005	9006
9050	9071	9073	9081	9135	9146	9147	9155	9156	9164	9165	9170	9171
9183	9184	9209	9230	9232	9240	9331	9345	9348	9349	9356	9357	9376
9377	9387	9395	9396	9405	9426	9428	9436					
2029	10386#											
1539#	2023#	2024#	2025#	2028#	2036	2173	2174	2215	2216	2268	2269	2306
2364	2365	2366	2393	2394	2446	2458	2459	2476	2487	2589	2590	2604
2605	2674	2686	2687	2704	2813	2851	2863	2864	2881	2907	2908	2931
2932	2977	2978	3003	3004	3018	3030	3031	3048	3057	3058	3152	3164
3165	3184	3212	3213	3214	3236	3248	3249	3268	3314	3323	3324	3338
3339	3387	3388	3400	3401	3431	3443	3444	3463	3508	3520	3521	3522
3549	3550	3591	3592	3602	3610	3611	3625	3637	3638	3657	3704	3715
3750	3751	3799	3811	3812	3831	3877	3885	3886	3914	3915	3935	3936
3957	3958	3972	3984	3985	4004	4051	4052	4053	4085	4086	4128	4129
4180	4192	4193	4212	4255	4256	4270	4271	4309	4310	4352	4353	4372
4373	4387	4399	4460	4419	4458	4470	4471	4472	4515	4527	4528	4547
4585	4594	4595	4609	4610	4627	4628	4642	4654	4655	4674	4717	4726
4727	4741	4742	4782	4783	4797	4809	4810	4829	4871	4883	4884	4885
4908	4909	4951	4963	4954	4983	5025	5048	5049	5055	5056	5082	5087
5088	5100	5101	5131	5143	5144	5163	5193	5207	5208	5218	5226	5227
5241	5253	5254	5273	5312	5340	5352	5353	5372	5418	5427	5428	5445
5446	5458	5459	5471	5472	5525	5537	5538	5556	5600	5612	5613	5614
5637	5638	5648	5670	5671	5692	5693	5707	5719	5720	5738	5786	5818
5819	5846	5847	5848	5915	5927	5928	5947	5957	5958	6014	6026	6027
6071	6072	6086	6087	6128	6129	6143	6155	6156	6175	6246	6278	6279
6306	6307	6308	6376	6388	6389	6408	6419	6420	6479	6491	6492	6536
6537	6551	6552	6594	6595	6609	6621	6622	6641	6705	6737	6738	6771
6783	6784	6802	6835	6836	6890	6920	6921	6953	6965	6966	6984	7014
7015	7052	7059	7068	7069	7070	7082	7083	7097	7109	7110	7129	7168
7172	7213	7225	7226	7245	7291	7346	7358	7359	7377	7382	7383	7441
7445	7446	7447	7481	7482	7496	7508	7509	7527	7620	7713	7725	7726
7743	7749	7750	7789	7790	7804	7816	7817	7834	7884	7885	7965	7966
7980	7992	7993	8010	8071	8083	8084	8101	8143	8154	8166	8167	8247

PORTA1 071477
 PORT8 001226

GO1

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
 DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 214
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0215

	8259	8260	8279	8289	8290	8347	8355	8376	8382	8383	8451	8463	8464	
	8483	8513	8554	8566	8590	8591	8612	8624	8625	8643	8693	8702	8703	
	8751	8763	8764	8782	8839	8848	8849	8859	8860	8909	8921	8922	8941	
	8985	8997	8998	8999	9055	9067	9068	9087	9139	9149	9152	9153	9160	
	9161	9180	9181	9191	9199	9200	9214	9226	9227	9245	9335	9342	9343	
	9351	9352	9360	9361	9366	9367	9379	9380	9410	9422	9423	9441		
PORTBI	071525	2035	10390#											
PORTC	001230	1540#	2042#	2043#	2044#	5033	5316							
PRO	= 000700	1171#												
PR1	= 000040	1172#												
PR2	= 000100	1173#												
PR3	= 000140	1174#												
PR4	= 000200	1175#												
PR5	= 000240	1176#												
PR6	= 000300	1177#												
PR7	= 000340	1178#												
PS	= 177776	1151#	1152	2060#	2346#	2416#	2433#	2574#	2644#	2661#				
PSEL	= 002000	1262#												
PSW	= 177776	1152#												
PTNBR	001240	1544#	2153#	2174#	2199#	2216#	2237#	2269#	2365#	2377#	2394#	2458#	2462#	2470#
		2476#	2487#	2593#	2605#	2622#	2686#	2690#	2698#	2704#	2715#	2837#	2863#	2867#
		2875#	2881#	2891#	2908#	2932#	3004#	3030#	3034#	3042#	3048#	3058#	3075#	3099#
		3129#	3164#	3168#	3178#	3184#	3213#	3248#	3252#	3262#	3268#	3327#	3339#	3356#
		3388#	3398#	3401#	3408#	3417#	3443#	3447#	3457#	3463#	3521#	3533#	3550#	3582#
		3592#	3595#	3602#	3611#	3637#	3641#	3651#	3657#	3742#	3751#	3763#	3785#	3811#
		3815#	3825#	3831#	3915#	3924#	3936#	3958#	3984#	3988#	3998#	4004#	4052#	4064#
		4086#	4103#	4129#	4146#	4166#	4192#	4196#	4206#	4212#	4259#	4271#	4293#	4310#
		4336#	4353#	4373#	4399#	4403#	4413#	4419#	4471#	4483#	4501#	4527#	4531#	4541#
		4547#	4579#	4610#	4628#	4654#	4658#	4668#	4674#	4730#	4742#	4755#	4783#	4809#
		4813#	4823#	4829#	4884#	4896#	4909#	4937#	4963#	4967#	4977#	4983#	5041#	5049#
		5056#	5088#	5098#	5101#	5108#	5117#	5143#	5147#	5157#	5163#	5176#	5198#	5208#
		5211#	5218#	5227#	5253#	5257#	5267#	5273#	5352#	5356#	5366#	5372#	5432#	5446#
		5456#	5459#	5466#	5472#	5489#	5511#	5537#	5541#	5551#	5556#	5614#	5628#	5638#
		5641#	5648#	5654#	5671#	5693#	5719#	5723#	5733#	5738#	5799#	5819#	5847#	5859#
		5901#	5927#	5931#	5941#	5947#	5958#	6027#	6047#	6075#	6087#	6129#	6155#	6159#
		6169#	6175#	6186#	6259#	6279#	6307#	6319#	6362#	6388#	6392#	6402#	6408#	6420#
		6492#	6512#	6540#	6552#	6595#	6621#	6625#	6635#	6641#	6653#	6718#	6738#	6783#
		6787#	6797#	6802#	6819#	6836#	6901#	6921#	6965#	6969#	6979#	6984#	7011#	7015#
		7069#	7083#	7109#	7113#	7123#	7129#	7185#	7199#	7225#	7229#	7239#	7245#	7296#
		7332#	7358#	7362#	7372#	7377#	7383#	7446#	7482#	7508#	7512#	7522#	7527#	7533#
		7699#	7725#	7729#	7737#	7743#	7750#	7790#	7816#	7820#	7828#	7834#	7966#	7992#
		7996#	8004#	8010#	8017#	8057#	8083#	8087#	8095#	8101#	8167#	8179#	8259#	8263#
		8273#	8279#	8290#	8371#	8383#	8463#	8467#	8477#	8483#	8494#	8591#	8624#	8628#
		8638#	8643#	8730#	8763#	8767#	8777#	8782#	8853#	8860#	8921#	8925#	8935#	8941#
		8999#	9006#	9067#	9071#	9081#	9087#	9153#	9156#	9161#	9165#	9171#	9181#	9184#
		9191#	9200#	9226#	9230#	9240#	9245#	9349#	9352#	9357#	9361#	9367#	9377#	9380#
		9387#	9396#	9422#	9426#	9436#	9441#	11077	11081	11086	11089	11092	11100	
PWRVEC=	000024	1243#												
ROCHR	= 104410	10161	10353#											
ROLIN	= 104411	10234	10354#											
ROOCT	= 104412	2017	2073	2099	10355#									
ROY	= 000200	1259#												
RELERR	001254	1550#	2437#	2467#	2665#	2695#	2842#	2872#	2887	3009#	3039#	3054	3143#	3173#
		3227#	3257#	3393#	3422#	3452#	3587#	3616#	3646#	3790#	3820#	3963#	3993#	4171#
		4201#	4378#	4408#	4506#	4536#	4633#	4663#	4788#	4818#	4942#	4972#	5093#	5122#
		5152#	5203#	5232#	5262#	5331#	5361#	5451#	5516#	5546#	5633#	5698#	5728#	5906#

H01

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 215
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0216

RELOK = 000001	5936*	6134*	6164*	6367*	6397*	6600*	6630*	6762*	6792*	6807	6944*	6974*	6989
	7088*	7118*	7204*	7234*	7337*	7367*	7487*	7517*	7704*	7734*	7795*	7825*	7971*
	8001*	8062*	8092*	8238*	8268*	8442*	8472*	8603*	8633*	8742*	8772*	8900*	8930*
	9046*	9076*	9176*	9205*	9235*	9372*	9401*	9431*					
	1956*	2468	2696	2873	3040	3107*	3174	3258	3453	3647	3821	3994	4202
	4409	4537	4664	4819	4973	5153	5263	5362	5547	5729	5937	6165	6398
	6631	6793	6975	7119	7235	7368	7518	7735	7826	8002	8093	8269	8473
	8634	8773	8931	9077	9236	9432							
RESREG= 104414	10357*												
RESVEC= 000010	1238*												
RMAS = 000016	1436*	5039*	5323*	6995*	6998	7002	7062	7072	7178	7188	9159*	9355*	
RMBR = 000004	1431*												
RMCS1 = 000000	1429*	2814*	2838*	2981*	3005*	3135*	3174*	3175*	3219*	3258*	3259*	3312*	3313*
	3316*	3317*	3357*	3371	3372	3377*	3389*	3418*	3453*	3454*	3506*	3507*	3510*
	3511*	3551*	3565	3566	3571*	3583*	3612*	3647*	3648*	3702*	3703*	3706*	3707*
	3721*	3723	3724	3743*	3778*	3786*	3821*	3822*	3875*	3876*	3879*	3880*	3894*
	3896	3897	3916*	3951*	3959*	3994*	3995*	4080*	4167*	4202*	4203*	4287*	4374*
	4409*	4410*	4456*	4457*	4460*	4461*	4469	4502*	4537*	4538*	4583*	4584*	4587*
	4588*	4596	4629*	4664*	4665*	4715*	4716*	4719*	4720*	4728	4761	4762	4784*
	4819*	4820*	4869*	4870*	4873*	4874*	4882	4915	4916	4938*	4973*	4974*	5023*
	5024*	5027*	5028*	5089*	5118*	5153*	5154*	5199*	5228*	5263*	5264*	5310*	5311*
	5314*	5315*	5362*	5363*	5416*	5417*	5420*	5421*	5447*	5512*	5547*	5548*	5598*
	5599*	5602*	5603*	5629*	5694*	5729*	5730*	5875*	5902*	5937*	5938*	6103*	6130*
	6165*	6166*	6336*	6363*	6398*	6399*	6569*	6596*	6631*	6632*	6793*	6794*	6975*
	6976*	7050*	7051*	7054*	7055*	7084*	7119*	7120*	7166*	7167*	7170*	7171*	7200*
	7235*	7236*	7289*	7290*	7293*	7294*	7302*	7333*	7368*	7369*	7439*	7440*	7443*
	7444*	7452*	7483*	7518*	7519*	7658*	7659*	7700*	7784*	7785*	7791*	7925*	7926*
	7967*	8051*	8052*	8058*	8141*	8142*	8145*	8146*	8269*	8270*	8311*	8312*	8345*
	8346*	8349*	8350*	8473*	8474*	8515*	8516*	8552*	8553*	8556*	8557*	8634*	8635*
	8691*	8692*	8695*	8696*	8773*	8774*	8837*	8838*	8841*	8842*	8865*	8931*	8932*
	8983*	8984*	8987*	8988*	9011*	9077*	9078*	9137*	9138*	9141*	9142*	9172*	9201*
	9236*	9237*	9265	9266	9271*	9282	9283	9288*	9333*	9334*	9337*	9338*	9368*
	9397*	9432*	9433*	9461	9462	9467*	9478	9479	9484*				
RMCS2 = 000010	1433*	2147*	2152*	2156	2157	2172*	2173*	2177	2178	2193*	2198*	2215*	2236*
	226*	2304*	2306*	2361*	2364*	2376*	2393*	2441*	2446*	2459*	2464*	2589*	2592*
	260*	2621*	2669*	2674*	2687*	2692*	2810*	2830*	2836*	2846*	2851*	2864*	2869*
	2890*	2907*	2931*	2977*	2997*	3003*	3013*	3018*	3031*	3036*	3057*	3074*	3098*
	3128*	3147*	3152*	3165*	3170*	3212*	3231*	3236*	3249*	3254*	3310*	3314*	3323*
	3326*	3338*	3355*	3387*	3397*	3400*	3416*	3426*	3431*	3444*	3449*	3504*	3508*
	3517*	3520*	3532*	3549*	3581*	3591*	3594*	3610*	3620*	3625*	3638*	3643*	3700*
	3704*	3712*	3740*	3741*	3750*	3762*	3784*	3794*	3799*	3812*	3817*	3873*	3877*
	3885*	3913*	3914*	3923*	3935*	3957*	3967*	3972*	3985*	3990*	4048*	4051*	4063*
	4085*	4102*	4123*	4128*	4145*	4165*	4175*	4180*	4193*	4198*	4255*	4258*	4270*
	4292*	4309*	4330*	4335*	4352*	4372*	4382*	4387*	4400*	4405*	4454*	4458*	4467*
	4470*	4482*	4500*	4510*	4515*	4528*	4533*	4581*	4585*	4594*	4597*	4609*	4627*
	4637*	4642*	4655*	4660*	4713*	4717*	4726*	4729*	4741*	4754*	4782*	4792*	4797*
	4810*	4815*	4867*	4871*	4880*	4883*	4895*	4908*	4936*	4946*	4951*	4964*	4969*
	5021*	5025*	5033*	5040*	5048*	5055*	5087*	5097*	5100*	5116*	5126*	5131*	5144*
	5149*	5175*	5197*	5207*	5210*	5226*	5236*	5241*	5254*	5259*	5308*	5312*	5316*
	5335*	5340*	5353*	5358*	5414*	5418*	5427*	5431*	5445*	5455*	5458*	5471*	5488*
	5510*	5520*	5525*	5538*	5543*	5596*	5600*	5609*	5613*	5627*	5637*	5640*	5653*
	5670*	5692*	5702*	5707*	5720*	5725*	5781*	5786*	5791*	5798*	5818*	5843*	5846*
	5858*	5900*	5910*	5915*	5928*	5933*	5957*	6009*	6014*	6019*	6026*	6046*	6071*
	6074*	6086*	6128*	6138*	6143*	6156*	6161*	6185*	6241*	6246*	6251*	6258*	6278*
	6303*	6306*	6318*	6361*	6371*	6376*	6389*	6394*	6419*	6474*	6479*	6484*	6491*
	6511*	6536*	6539*	6551*	6594*	6604*	6609*	6622*	6627*	6652*	6700*	6705*	6710*

RMDA = 000006
RMD8 = 000022
RMDC = 000034
RMD51 = 000012

6717*	6737*	6766*	6771*	6784*	6789*	6813*	6818*	6835*	6885*	6890*	6895*	6900*	
6920*	6948*	6953*	6966*	6971*	7010*	7014*	7048*	7052*	7056*	7059*	7068*	7082*	
7092*	7097*	7110*	7115*	7164*	7168*	7172*	7175*	7184*	7198*	7208*	7213*	7226*	
7231*	7287*	7291*	7295*	7331*	7341*	7346*	7359*	7364*	7382*	7437*	7441*	7445*	
7481*	7491*	7496*	7509*	7514*	7532*	7617*	7698*	7708*	7713*	7726*	7731*	7749*	
7789*	7799*	7804*	7817*	7822*	7884*	7965*	7975*	7980*	7993*	7998*	8016*	8056*	
8066*	8071*	8084*	8089*	8139*	8143*	8151*	8166*	8178*	8242*	8247*	8260*	8265*	
8289*	8309*	8343*	8347*	8355*	8370*	8382*	8446*	8451*	8464*	8469*	8493*	8513*	
8550*	8554*	8563*	8590*	8607*	8612*	8625*	8630*	8689*	8693*	8702*	8729*	8746*	
8751*	8764*	8769*	8835*	8839*	0848*	8852*	8859*	8904*	8909*	8922*	8927*	8981*	
8985*	8994*	8998*	9005*	9050*	9055*	9068*	9073*	9135*	9139*	9146*	9152*	9155*	
9160*	9164*	9170*	9180*	9183*	9199*	9209*	9214*	9227*	9232*	9331*	9335*	9342*	
9348*	9351*	9356*	9360*	9366*	9376*	9379*	9395*	9405*	9410*	9423*	9428*		
1432*	2363*	2403	2499	2591*	2631	2727							
1438*													
1443*	2398	2524	2626	2752									
1434*	2154	2175	2239	2240	2254	2255	2271	2272	2286	2287	2367	2369	
2378	2410	2438	2442	2447	2595	2597	2606	2638	2666	2670	2675	2812*	
2816	2817	2843	2847	2852	2893	2894	2910	2911	2933	2979*	2983	2984	
3010	3014	3019	3060	3061	3077	3078	3100	3144	3148	3153	3228	3232	
3237	3311*	3315*	3325*	3329	3331	3340	3363	3364	3395	3399	3402	3423	
3427	3432	3505*	3509*	3519*	3523	3525	3534	3557	3558	3589	3593	3596	
3617	3621	3626	3701*	3705*	3714*	3753	3754	3791	3795	3800	3874*	3878*	
3887*	3926	3927	3964	3968	3973	4050*	4054	4056	4065	4088	4089	4105	
4106	4131	4132	4148	4149	4172	4176	4181	4257*	4261	4263	4272	4295	
4296	4312	4313	4338	4339	4355	4356	4379	4383	4388	4455*	4459*	4473	
4475	4484	4507	4511	4516	4582*	4586*	4600	4602	4611	4634	4638	4643	
4714*	4718*	4732	4734	4743	4789	4793	4798	4868*	4872*	4886	4888	4897	
4943	4947	4952	5022*	5026*	5046	5050	5057	5068	5069	5095	5099	5102	
5123	5127	5132	5178	5179	5205	5209	5212	5233	5237	5242	5309*	5313*	
5332	5336	5341	5415*	5419*	5429*	5437*	5453	5457	5460	5474	5475	5491	
5492	5517	5521	5526	5597*	5601*	5611*	5619*	5635	5639	5642	5656	5657	
5673	5674	5699	5703	5708	5782	5787	5792	5801	5802	5821	5822	5845*	
5849	5851	5860	5881	5882	5907	5911	5916	5960	5961	6010	6015	6020	
6029	6030	6049	6050	6073*	6077	6079	6088	6109	6110	6135	6139	6144	
6188	6189	6242	6247	6252	6261	6262	6281	6282	6305*	6309	6311	6320	
6342	6343	6368	6372	6377	6422	6423	6475	6480	6485	6494	6495	6514	
6515	6538*	6542	6544	6553	6575	6576	6601	6605	6610	6655	6656	6701	
6706	6711	6720	6721	6740	6741	6763	6767	6772	6821	6822	6838	6839	
6886	6891	6896	6903	6904	6923	6924	6945	6949	6954	7012	7016	7049*	
7053*	7060	7071*	7089	7093	7098	7165*	7169*	7176	7187*	7205	7209	7214	
7288*	7292*	7307	7314	7315	7338	7342	7347	7389	7390	7438*	7442*	7457	
7464	7465	7488	7492	7497	7539	7540	7627	7628	7642	7643	7663	7664	
7678	7679	7705	7709	7714	7752	7753	7767	7768	7796	7800	7805	7894	
7895	7909	7910	7930	7931	7945	7946	7972	7976	7981	8019	8020	8034	
8035	8063	8067	8072	8140*	8144*	8153*	8172	8185	8186	8218	8219	8239	
8243	8248	8292	8293	8310*	8344*	8348*	8357*	8376	8389	8390	8422	8423	
8443	8447	8452	8496	8497	8514*	8551*	8555*	8565*	8589	8592	8604	8608	
8613	8690*	8694*	8704*	8728	8731	8743	8747	8752	8836*	8840*	8850*	8858*	
8876	8877	8901	8905	8910	8982*	8986*	8996*	9004*	9022	9023	9047	9051	
9056	9136*	9140*	9148*	9178	9182	9185	9206	9210	9215	9332*	9336*	9344*	
9374	9378	9381	9402	9406	9411								
1440*	2201	2202	2218	2219	2402	2504	2630	2732					
RMEC1 = 000044	1446*	2396	2534	2624	2762								
RMEC2 = 000046	1447*	2395	2539	2623	2767								
RMER1 = 000014	1435*	2405	2489	2633	2717	3720*	3765	3766	3893*	3938	3939	5784*	5785*

NO1

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 221
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0222

	8576*	8577	8584*	8594	8715*	8716	8723*	8733	8870*	8891	9016*	9037	9581
	9583*	9585*											
WCE = 040000	1289#												
WCF = 000040	1337#												
WLF = 004000	1343#												
WRL = 004000	1324#	2443	2448	2671	2676	2848	2853	3015	3020	3149	3154	3233	3238
	3428	3433	3622	3627	3796	3801	3969	3974	4177	4182	4384	4389	4512
	4517	4639	4644	4794	4799	4948	4953	5128	5133	5238	5243	5337	5342
	5522	5527	5704	5709	5912	5917	6140	6145	6373	6378	6606	6611	6768
	6773	6950	6955	7094	7099	7210	7215	7343	7348	7493	7498	7710	7715
	7801	7806	7977	7982	8068	8073	8244	8249	8448	8453	8609	8614	8748
	8753	8906	8911	9052	9057	9211	9216	9407	9412				
SAUTO8 001134	1503#	2013*	10066	10214									
SBOADR 001122	1498#	2157*	2158*	2178*	2179*	2202*	2203*	2219*	2220*	2240*	2241*	2255*	2256*
	2272*	2273*	2287*	2288*	2368*	2369*	2438*	2439*	2488*	2489*	2493*	2494*	2498*
	2499*	2503*	2504*	2506*	2509*	2513*	2514*	2518*	2514*	2523*	2524*	2528*	2529*
	2533*	2534*	2538*	2539*	2596*	2597*	2666*	2667*	2716*	2717*	2721*	2722*	2726*
	2727*	2731*	2732*	2736*	2737*	2741*	2742*	2746*	2747*	2751*	2752*	2756*	2757*
	2761*	2762*	2766*	2767*	2817*	2818*	2843*	2844*	2894*	2895*	2911*	2912*	2984*
	2985*	3010*	3011*	3061*	3062*	3078*	3079*	3144*	3145*	3228*	3229*	3330*	3331*
	3364*	3365*	3372*	3373*	3395*	3396*	3423*	3424*	3524*	3525*	3558*	3559*	3566*
	3567*	3589*	3590*	3617*	3618*	3724*	3725*	3754*	3755*	3766*	3767*	3791*	3792*
	3897*	3898*	3927*	3928*	3939*	3940*	3964*	3965*	4055*	4056*	4089*	4090*	4106*
	4107*	4132*	4133*	4149*	4150*	4172*	4173*	4262*	4263*	4296*	4297*	4313*	4314*
	4339*	4340*	4356*	4357*	4379*	4380*	4474*	4475*	4507*	4508*	4601*	4602*	4634*
	4635*	4733*	4734*	4762*	4763*	4789*	4790*	4887*	4888*	4916*	4917*	4943*	4944*
	5069*	5070*	5095*	5096*	5123*	5124*	5179*	5180*	5205*	5206*	5233*	5234*	5332*
	5333*	5453*	5454*	5475*	5476*	5492*	5493*	5517*	5518*	5635*	5636*	5657*	5658*
	5674*	5675*	5699*	5700*	5802*	5803*	5822*	5823*	5850*	5851*	5882*	5883*	5907*
	5908*	5961*	5962*	6030*	6031*	6050*	6051*	6078*	6079*	6110*	6111*	6135*	6136*
	6189*	6190*	6262*	6263*	6282*	6283*	6310*	6311*	6343*	6344*	6368*	6369*	6423*
	6424*	6495*	6496*	6515*	6516*	6543*	6544*	6576*	6577*	6601*	6602*	6656*	6657*
	6721*	6722*	6741*	6742*	6763*	6764*	6822*	6823*	6839*	6840*	6904*	6905*	6924*
	6925*	6945*	6946*	7001*	7002*	7062*	7063*	7089*	7090*	7178*	7179*	7205*	7206*
	7315*	7316*	7338*	7339*	7390*	7391*	7465*	7466*	7488*	7489*	7540*	7541*	7628*
	7629*	7643*	7644*	7664*	7665*	7679*	7680*	7705*	7706*	7753*	7754*	7768*	7769*
	7796*	7797*	7895*	7896*	7910*	7911*	7931*	7932*	7946*	7947*	7972*	7973*	8020*
	8021*	8035*	8036*	8063*	8064*	8186*	8187*	8205*	8206*	8219*	8220*	8239*	8240*
	8293*	8294*	8390*	8391*	8409*	8410*	8423*	8424*	8443*	8444*	8497*	8498*	8604*
	8605*	8743*	8744*	8877*	8878*	8901*	8902*	9023*	9024*	9047*	9048*	9178*	9179*
	9206*	9207*	9258*	9259*	9266*	9267*	9275*	9276*	9283*	9284*	9374*	9375*	9402*
	9403*	9454*	9455*	9462*	9463*	9471*	9472*	9479*	9480*	11077	11079	11081	11084
	11086	11089	11094										
SBDAT 001126	1500#	2156*	2160	2164	2177*	2181	2185	2201*	2205	2209	2218*	2222	2226
	2239*	2243	2247	2254*	2258	2262	2271*	2275	2279	2286*	2290	2294	2307*
	2308	2367*	2371	2378*	2379*	2383	2457*	2463*	2469*	2471*	2472	2475*	2477*
	2478	2490*	2495*	2500*	2505*	2510*	2515*	2520*	2525*	2530*	2535*	2540*	2595*
	2599	2606*	2607*	2611	2685*	2691*	2697*	2699*	2700	2703*	2705*	2706	2718*
	2723*	2728*	2733*	2738*	2743*	2748*	2753*	2758*	2763*	2768*	2816*	2820	2824
	2862*	2868*	2874*	2876*	2877	2880*	2882*	2383	2893*	2897	2901	2910*	2914
	2918	2983*	2987	2991	3029*	3035*	3041*	3043*	3044	3047*	3049*	3050	3060*
	3064	3068	3077*	3081	3085	3163*	3169*	3177*	3179*	3180	3183*	3185*	3186
	3247*	3253*	3261*	3263*	3264	3267*	3269*	3270	3329*	3333	3340*	3341*	3345
	3363*	3367	3371*	3374	3402*	3407*	3409	3442*	3448*	3456*	3458*	3459	3462*
	3464*	3465	3523*	3527	3534*	3535*	3539	3557*	3561	3565*	3568	3596*	3601*
	3603	3636*	3642*	3650*	3652*	3653	3656*	3658*	3659	3723*	3727	3731	3753*

3757	3765*	3769	3810*	3816*	3824*	3826*	3827	3830*	3832*	3833	3896*	3900
3904	3926*	3930	3938*	3942	3983*	3989*	3997*	3999*	4000	4003*	4005*	4006
4054*	4058	4065*	4066*	4070	4088*	4092	4096	4105*	4109	4113	4131*	4135
4139	4148*	4152	4156	4191*	4197*	4205*	4207*	4208	4211*	4213*	4214	4261*
4265	4272*	4273*	4277	4295*	4299	4303	4312*	4316	4320	4338*	4342	4346
4355*	4359	4363	4398*	4404*	4412*	4414*	4415	4418*	4420*	4421	4473*	4477
4484*	4485*	4489	4526*	4532*	4540*	4542*	4543	4546*	4548*	4549	4600*	4604
4611*	4612*	4616	4653*	4659*	4667*	4669*	4670	4673*	4675*	4676	4732*	4736
4743*	4744*	4748	4761*	4765	4769	4808*	4814*	4823*	4824*	4825	4828*	4830*
4831	4886*	4890	4897*	4898*	4902	4915*	4919	4923	4962*	4968*	4976*	4978*
4979	4982*	4984*	4985	5068*	5072	5076	5102*	5107*	5109	5142*	5148*	5156*
5158*	5159	5162*	5164*	5165	5178*	5182	5186	5212*	5217*	5219	5252*	5258*
5266*	5268*	5269	5272*	5274*	5275	5351*	5357*	5365*	5367*	5368	5371*	5373*
5374	5460*	5465*	5467	5474*	5478	5482	5491*	5495	5499	5536*	5542*	5550*
5552	5555*	5557	5642*	5647*	5649	5656*	5660	5664	5673*	5677	5681	5718*
5724*	5732*	5734	5737*	5739	5801*	5805	5809	5821*	5825	5829	5844*	5853
5860*	5861*	5865	5881*	5885	5889	5926*	5932*	5940*	5942*	5943	5946*	5948*
5949	5960*	5964	5968	6029*	6033	6037	6049*	6053	6057	6077*	6081	6088*
6089*	6093	6109*	6113	6117	6154*	6160*	6168*	6170*	6171	6174*	6176*	6177
6188*	6192	6196	6261*	6265	6269	6281*	6285	6289	6309*	6313	6320*	6321*
6325	6342*	6346	6350	6387*	6393*	6401*	6403*	6404	6407*	6409*	6410	6422*
6426	6430	6494*	6498	6502	6514*	6518	6522	6542*	6546	6553*	6554*	6558
6575*	6579	6583	6620*	6626*	6634*	6636*	6637	6640*	6642*	6643	6655*	6659
6663	6720*	6724	6728	6740*	6744	6748	6782*	6788*	6796*	6798	6801*	6803
6821*	6825	6829	6838*	6842	6846	6903*	6907	6911	6923*	6927	6931	6964*
6970*	6978*	6980	6983*	6985	6998*	6999	7003	7072*	7073	7108*	7114*	7122*
7124*	7125	7128*	7130*	7131	7188*	7189	7224*	7230*	7238*	7240*	7241	7244*
7246*	7247	7314*	7318	7322	7357*	7363*	7371*	7373	7376*	7378	7389*	7393
7397	7464*	7468	7472	7507*	7513*	7521*	7523	7526*	7528	7539*	7543	7547
7627*	7631	7635	7642*	7646	7650	7663*	7667	7671	7678*	7682	7686	7724*
7730*	7736*	7738*	7739	7742*	7744*	7745	7752*	7756	7760	7767*	7771	7775
7815*	7821*	7827*	7829*	7830*	7833*	7835*	7836	7894*	7898	7902	7909*	7913
7917	7930*	7934	7938	7945*	7949	7953	7991*	7997*	8003*	8005*	8006	8009*
8011*	8012	8019*	8023	8027	8034*	8038	8042	8082*	8088*	8094*	8096*	8097
8100*	8102*	8103	8185*	8189	8193	8204*	8208	8218*	8222	8226	8258*	8264*
8272*	8274*	8275	8278*	8280*	8281	8292*	8296	8300	8389*	8393	8397	8408*
8412	8422*	8426	8430	8462*	8468*	8476*	8478*	8479	8482*	8484*	8485	8496*
8500	8504	8623*	8629*	8637*	8679	8642*	8644	8762*	8768*	8776*	8778	8781*
8783	8876*	8880	8920*	8926*	8934*	8936*	8937	8940*	8942*	8943	9022*	9026
9066*	9072*	9080*	9082*	9083	9086*	9088*	9099	9185*	9190*	9192	9225*	9231*
9239*	9241	9244*	9246	9257*	9261	9265*	9268	9274*	9278	9282*	9285	9381*
9386*	9388	9421*	9427*	9435*	9437	9440*	9442	9453*	9457	9461*	9464	9470*
9474	9478*	9481	11077	11079	11081	11084	11086	11089	11094			
1526*	9686	9706	10019	10076								
9798*	9808*	9815	9824*	9829*								
10052*	10352											
1486*	1967	1968	1976	1980	1981	1982						
1518*	1519*											
1518*	1519*											
1516*	1518											
1519*	1520*	1521*	1522*	1523*	1524*							
10073	10209*											
10090	10183	10208*										
1528*	2041	2061	2097	9529	9694	9706	9714	9733	9738	9758	9797	9832
10101	10188	10208	10268									
9933	9967	9975*										

SBELL 001202
 SCHARC 067156
 SCKSWR 070106
 SCMTAG 001100
 SCH1 = 000001
 SCH2 = 000002
 SCH3 = 000001
 SCH4 = 000005
 SCNTLG 070750
 SCNTLU 070743
 SCRLF 001207
 SOBLK 067624

E02

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 225
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0226

SSVLAD 066370
SSVPC = 000200
SSMR = 166000

SSWRMK= 000000
\$TIMES 001176

\$TKB 001146
\$TKCNT 067634
\$TKINT 067644
\$TKQEN= 067643
\$TKQIN 067636
\$TKQOU 067640
\$TKQSR 067642
\$TKS 001144
\$TKSRV 067714
\$TMPD 001164

9631	9656#													
1463#	1468													
1115#	1126	1131	1132	1133	1134	1135	1136	1137	1524	1525	1526	1981		
1982	1984	1985	2145	2344	2572	2796	2963	3126	3210	3305	3499	3695		
3868	4041	4248	4449	4576	4708	4862	5016	5303	5409	5591	5775	6003		
6234	6467	6694	6879	7043	7159	7282	7432	7607	7874	8134	8338	8545		
8684	8830	8976	9130	9326	9490	9498	9506	9533	9539	9541	9613	9614		
9615	9616	9617	9622	9634	9636	9637	9638	9645	9646	9647	9658	9661		
9664	9665#	9671	9672	9673	9674	9675	9684	9691	9696	9700	9706			
9617														
1524#	1981*	2145*	2344*	2572*	2796*	2963*	3126*	3210*	3305*	3499*	3695*	3868*		
4041*	4248*	4449*	4576*	4708*	4862*	5016*	5303*	5409*	5591*	5775*	6003*	6234*		
6467*	6694*	6879*	7043*	7159*	7282*	7432*	7607*	7874*	8134*	8338*	8545*	8684*		
8830*	8976*	9130*	9326*	9506*	9645*	9652	9655*	9664						
1509#	9979	10000	10009	10028	10056	10083								
9980#	9995*	10017	10034*	10140	10142*									
2005	2066	9995#	10069											
9984#	10042	10145												
9981#	9996*	9997	10040*	10041*	10042	10044*								
9982#	9997*	10143	10144*	10145	10147*									
9983#	9996	10044	10147											
1508#	9979	10001*	10024*	10026	10032*	10054	10070*	10080	10104*					
9998	10009#													
1519#	2160*	2.61*	2162	2181*	2182*	2183	2205*	2206*	2207	2222*	2223*	2224		
2243*	2244*	2245	2258*	2259*	2260	2275*	2276*	2277	2290*	2291*	2292	2383*		
2384*	2385	2444*	2445*	2451	2453	2460	2611*	2612*	2613	2672*	2673*	2679		
2681	2688	2820*	2821*	2822	2849*	2850*	2856	2858	2865	2897*	2898*	2899		
2914*	2915*	2916	2987*	2988*	2989	3016*	3017*	3023	3025	3032	3064*	3065*		
3066	3081*	3082*	3083	3150*	3151*	3157	3159	3166	3234*	3235*	3241	3243		
3250	3345*	3346*	3347	3399*	3404	3407	3429*	3430*	3436	3438	3445	3539*		
3540*	3541	3593*	3598	3601	3623*	3624*	3630	3632	3639	3727*	3728*	3729		
3797*	3798*	3804	3806	3813	3900*	3901*	3902	3970*	3971*	3977	3979	3986		
4070*	4071*	4072	4092*	4093*	4094	4109*	4110*	4111	4135*	4136*	4137	4152*		
4153*	4154	4178*	4179*	4185	4187	4194	4277*	4278*	4279	4299*	4300*	4301		
4316*	4317*	4318	4342*	4343*	4344	4359*	4360*	4361	4385*	4386*	4392	4394		
4401	4489*	4490*	4491	4513*	4514*	4520	4522	4529	4616*	4617*	4618	4640*		
4641*	4647	4649	4656	4748*	4749*	4750	4765*	4766*	4767	4795*	4796*	4802		
4804	4811	4902*	4903*	4904	4919*	4920*	4921	4949*	4950*	4956	4958	4965		
5072*	5073*	5074	5099*	5104	5107	5129*	5130*	5136	5138	5145	5182*	5183*		
5184	5209*	5214	5217	5239*	5240*	5246	5248	5255	5321*	5322*	5323	5338*		
5339*	5345	5347	5354	5457*	5462	5465	5478*	5479*	5480	5495*	5496*	5497		
5523*	5524*	5530	5532	5539	5639*	5644	5647	5660*	5661*	5662	5677*	5678*		
5679	5705*	5706*	5712	5714	5721	5805*	5806*	5807	5825*	5826*	5827	5865*		
5866*	5867	5885*	5886*	5887	5913*	5914*	5920	5922	5929	5964*	5965*	5966		
6033*	6034*	6035	6053*	6054*	6055	6093*	6094*	6095	6113*	6114*	6115	6141*		
6142*	6148	6150	6157	6192*	6193*	6194	6265*	6266*	6267	6285*	6286*	6287		
6325*	6326*	6327	6346*	6347*	6348	6374*	6375*	6381	6383	6390	6426*	6427*		
6428	6498*	6499*	6500	6518*	6519*	6520	6558*	6559*	6560	6579*	6580*	6581		
6607*	6608*	6614	6616	6623	6659*	6660*	6661	6724*	6725*	6726	6744*	6745*		
6746	6769*	6770*	6776	6778	6785	6825*	6826*	6827	6842*	6843*	6844	6907*		
6908*	6909	6927*	6928*	6929	6951*	6952*	6958	6960	6967	7073*	7074*	7075		
7095*	7096*	7102	7104	7111	7189*	7190*	7191	7211*	7212*	7218	7220	7227		
7318*	7319*	7320	7344*	7345*	7351	7353	7360	7393*	7394*	7395	7468*	7469*		
7470	7494*	7495*	7501	7503	7510	7543*	7544*	7545	7631*	7632*	7633	7646*		
7647*	7648	7667*	7668*	7669	7682*	7683*	7684	7711*	7712*	7718	7720	7727		
7756*	7757*	7758	7771*	7772*	7773	7802*	7803*	7809	7811	7818	7898*	7899*		

H02

MD-11-DZRMG-A, RMO3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 228
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0229

\$XTSTR 066214
\$\$GET4= 000000
\$OFILL 067405
\$4OCAT= ***** U
= 101002

9625#													
9533#													
9858#	9862#	9872	9907#										
9622	9693												
1451#	1455#	1463	1464#	1466#	1468#	1485#	1530	1971	1984	1985	2070	2071	
2171	2192	2491	2496	2501	2506	2511	2516	2521	2526	2531	2536	2541	
2719	2724	2729	2734	2739	2744	2749	2754	2759	2764	2769	2888	3055	
3380	3574	3738	3911	5058	5816	5836	6044	6064	6276	6296	6509	6529	
6735	6755	6808	6918	6938	6990	7308	7458	8655	8794	8886	8892	9032	
9038	9503	9518#	9541	9542#	9664	9665	9706	9761#	9832	9975#	9979	9983#	
9984	9985#	10207#	10208	10214#	10268	11075#							

LL	9132#	9328													
LL1	9132#	9251	9447												
MM	7284#	7434													
MMO	6696#														
MM1	5777#	6005													
MORETA	1478#	1531													
MSG	2120#	2125	2316#	2319	2544#	2547	2772#	2775	2939#	2942	3107#	3110	3191#	3194	3276#
	3279	3470#	3473	3665#	3668	3838#	3841	4012#	4015	4219#	4222	4427#	4430	4554#	4557
	4682#	4685	4836#	4839	4991#	4994	5281#	5284	5380#	5383	5562#	5565	5746#	5749	5975#
	5978	6205#	6208	6438#	6441	6670#	6673	6855#	6858	7020#	7023	7136#	7139	7255#	7258
	7406#	7409	7557#	7559	7567#	7569	7577#	7579	7844#	7846	8110#	8113	8314#	8317	8519#
	8522	8658#	8661	8799#	8802	8948#	8951	9095#	9098	9291#	9294				
MULT	1249#														
NEUTRA	1115#	2434	2662	2839	3006	3140	3224	3419	3613	3787	3960	4168	4375	4503	4630
	4785	4939	5119	5229	5328	5513	5695	5903	6131	6364	6597	6759	6941	7085	7201
	7334	7484	7701	7792	7968	8059	8235	8439	8600	8739	8897	9043	9202	9398	
NEWTST	1249#	2123	2317	2545	2773	2940	3108	3192	3277	3471	3666	3839#	4013	4220	4428
	4555	4683	4837	4992	5282	5381	5563	5747	5976	6206	6439	6671	6856	7021	7137
	7256	7407	7577	7844	8111	8315	8520	8659	8800	8949	9096	9292	9488		
NN	8832#	8978													
OO	4043#	4250													
POP	1249#	9962	10257	10306											
PUSH	1249#	9921	10231	10286											
RELEAS	1115#	2833	3000	3384	3413	3578	3607	3781	3954	4162	4369	4497	4624	4779	4933
	5084	5113	5194	5223	5142	5507	5624	5689	5897	6125	6358	6591	7078	7194	7328
	7478	7695	7786	7962	8053	9167	9196	9363	9392						
REPORT	1249#														
RR	7045#	7161													
SCOPE	1144#	2314	2543	2771	2938	3105	3190	3274	3469	3663	3837	4010	4218	4425	4553
	4680	4835	4989	5279	5378	5561	5743	5974	6202	6436	6669	6852	7018	7135	7251
	7405	7555	7841	8108	8313	8517	8657	8796	8947	9093	9250	9446	9489		
SEIZE	1115#	2358	2586	2807	2974	3320	3514	3709	3882	4045	4252	4464	4591	4723	4877
	5424	5606	5840	6068	6300	6533	7614	7881	8148	8352	8560	8699	8845	8991	9143
	9339														
SELECT	1115#	2152	2173	2198	2215	2236	2268	2364	2375	2393	2592	2603	2621	2836	2890
	2907	2930	3003	3057	3074	3097	3128	3212	3326	3337	3355	3387	3397	3400	3416
	3520	3531	3549	3581	3591	3594	3610	3741	3749	3762	3784	3914	3922	3935	3957
	4051	4062	4085	4102	4128	4145	4165	4258	4269	4292	4309	4335	4352	4372	4470
	4481	4500	4597	4608	4627	4729	4740	4754	4782	4883	4894	4908	4936	5040	5048
	5054	5087	5097	5100	5116	5174	5197	5207	5210	5226	5431	5445	5455	5458	5471
	5488	5510	5613	5627	5637	5640	5653	5670	5692	5798	5818	5846	5857	5900	5957
	6026	6046	6074	6085	6128	6185	6258	6278	6306	6317	6361	5419	6491	6511	6539
	6550	6594	6652	6717	6737	6818	6835	6900	6920	7010	7014	7068	7082	7184	7198
	7295	7331	7382	7445	7481	7532	7698	7749	7789	7965	8016	8056	8166	8177	8289
	8370	8381	8493	8590	8729	8852	8859	8998	9005	9152	9155	9160	9164	9170	9180
	9183	9199	9348	9351	9356	9360	9366	9376	9379	9395					
SETATA	1115#	5778	6006	6238	6471	6697	6882								
SETPRI	1249#	10136													
SETTRA	10336#	10345	10346	10347	10348	10350	10352	10353	10354	10355	10356	10357			
SETUP	1249#	1966													
SKIP	1249#														
SLASH	1249#														
SPACE	1249#														
STARS	1249#	1250	1254	1296	1300	1423	1427	1461	1481	1530	1563	1567	1956	1960	2114
	2118	2123	2135	2149	2195	2233	2301	2317	2334	2351	2357	2390	2407	2423	2430
	2483	2545	2562	2579	2585	2618	2635	2651	2658	2711	2773	2786	2799	2805	2832

L02

MD-11-DZRMG-A, RMD3 DUAL PORT LOGIC TEST - PART 1
DZRMGA.P11 01-AUG-77 10:58

MACY11 30(1046) 01-AUG-77 11:02 PAGE 233
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0233

.SSCOP 1115# 9607
.STRAP 1115# 10313
.STYPD 1115# 9909
.STYPE 1115# 9762
.STYPO 1115# 9832

. ABS. 101002 000

ERRORS DETECTED: 0

DSKZ:DZRMGA.BIN, DSKZ:DZRMGA.SEQ/DOC/SOL/CRF=DSKM:DZRMGA.P11
RUN-TIME: 38 40 4 SECONDS
RUN-TIME RATIO: 491/83=5.8
CORE USED: 32K (63 PAGES)

DOCUMENT PAGES: 233