

RP04/05/06

DUAL CONTROLLER LOGIC
MD-11-DZRJE-A

EP-DZRJE-A-DL-A

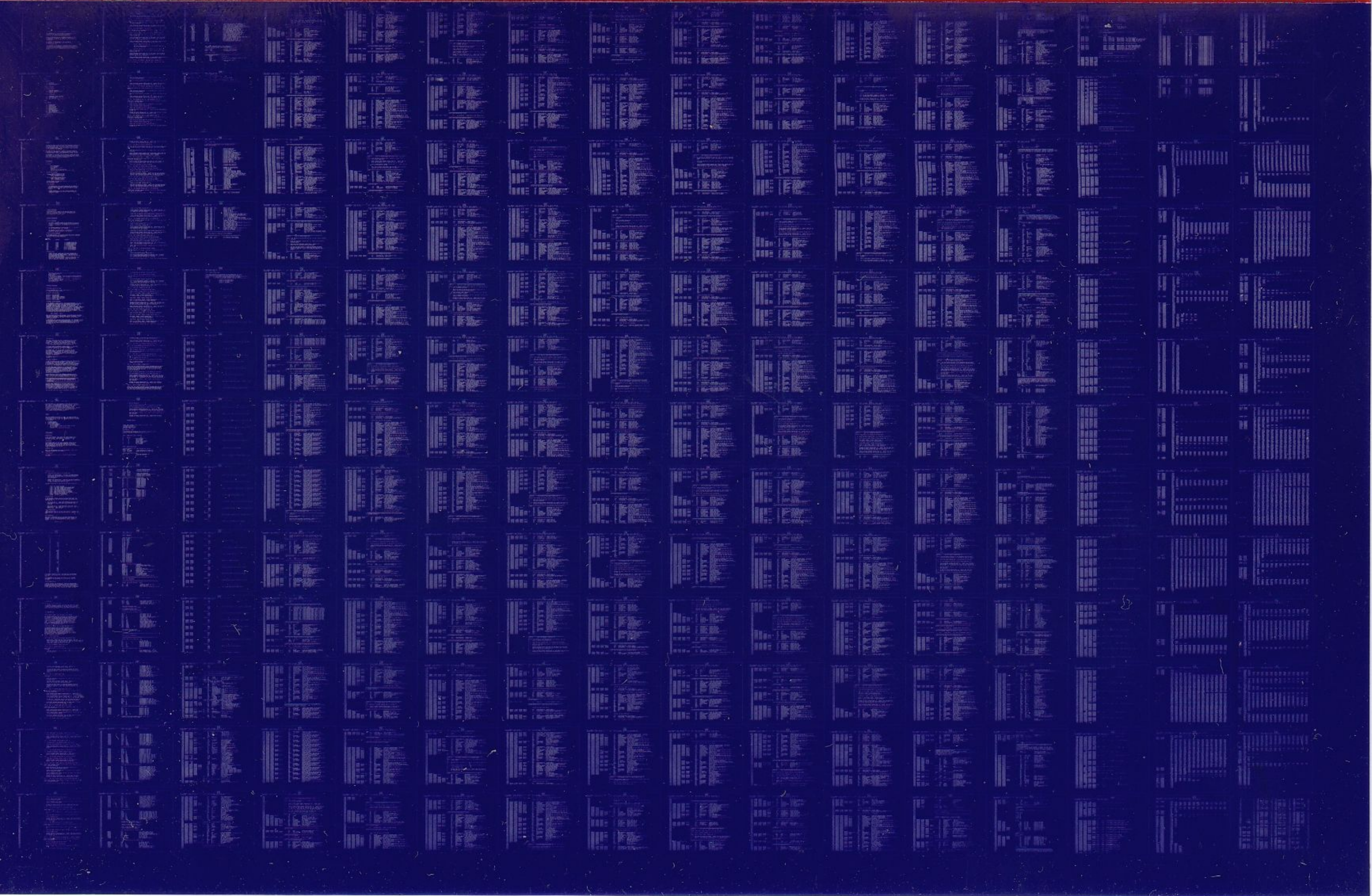
NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 2

MADE IN USA



RP04/05/06

DUAL CONTROLLER LOGIC
MD-11-DZRJE-A

EP-DZRJE-A-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 2 OF 2

MADE IN USA

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 PRELIMINARY 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 STACK POINTER
 - 7.5 SUBROUTINE CALLS
 - 7.6 REQUIRED TESTS
 - 7.7 DISK SURFACE USAGE
 - 7.8 TEST ITERATIONS
 - 7.9 LOOP ON ERROR OPTION
 - 7.10 SPECIAL M7775 'DP' BOARD TESTS
- 8. TEST DESCRIPTIONS
- 9. PROGRAM LISTING

MD-11-DZRJE-A. DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 2
DZRJEA.CMB 02-NOV-76 18:40

11001
11002
11003
11004
11005
11006
11007
11008
11009
11010
11011
11012
11013
11014
11015
11016
11017
11018
11019
11020
11021
11022
11023
11024
11025
11026
11027
11028
11029
11030
11031
11032
11033
11034
11035
11036
11037
11038
11039
11040
11041
11042
11043

1. ABSTRACT

THE RP04/5/6 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RP04/5/6 DUAL CONTROLLER LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL CONTROLLER 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 CONTROLLER LOGIC TO BE TESTED FROM ONE PDP-11/RH11 OR RH70.

THIS PROGRAM IS THE FIRST PART OF THE DUAL CONTROLLER OPTION LOGIC TEST. ALL OF THE DUAL CONTROLLER OPTION LOGIC, EXCEPT THE LOGIC ASSOCIATED WITH THE UNLOAD COMMAND AND THE CONTROLLER SELECT SWITCH, IS TESTED BY THIS PROGRAM.

2. REQUIREMENTS

2.1 EQUIPMENT

- PDP-11 PROCESSOR
- 16K OF MEMORY
- KW11-L OR KW11-P CLOCK
- TELETYPE
- RH11 OR RH70 WITH AN RP04/5/6
- RP04/5/6 DUAL CONTROLLER OPTION TEST CABLE

2.2 PRELIMINARY PROGRAMS

RP04/5/6 DISKLESS CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJG)
PART 2 (MAINDEC-11-DZRJH)

RP04/5/6 FUNCTIONAL CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJI)
PART 2 (MAINDEC-11-DZRJJ)

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

2.3 OTHER PROGRAMS

A. THE OPERATION OF THE UNLOAD COMMAND AND THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH ARE TESTED BY THE RP04/5/6 DUAL CONTROLLER LOGIC TEST, PART 2 (MAINDEC-11-DZRJF).

B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RP04/5/6 MULTIDRIVE EXERCISER PROGRAM (MAINDEC-11-DZRJD).

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
2003. 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.

4. STARTING PROCEDURES4.1 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 CONTROLLER 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 'CONTROLLER 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))

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256

- INTO THE SWITCH REGISTER (OR THE 'SOFTWARE' SWITCH REGISTER, REFER TO SECTION 5.2).
- E. PRESS START.
 - F. ENTER THE DRIVE NUMBER. (THIS MUST BE THE NUMBER DISPLAYED BY THE DRIVE, IF AN RPO4, OR THE NUMBER OF THE ADDRESS PLUG IF THE DRIVE IS AN RPO5/6).
 - G. ENTER THE NUMBER OF THE TEST TO BE RUN. ('CARRIAGE RETURN' OR '0' WILL RUN ALL TESTS.)
 - H. THE PROGRAM MAY BE STOPPED AT ANY TIME AND RESTARTED FROM LOCATION 204.

5. OPERATING PROCEDURES

5.1 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 RPO4/5/6 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.

257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312

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 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' (↑U).

5.4 TEST CABLE CONNECTION

TO TEST THE RPO4/5/6 DUAL CONTROLLER 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 RPO4 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 (THE ADDRESS SELECTED BY THE SWITCHES ON THE 'DP' BOARD - MODULE M7775 FOR RPO4'S, OR BY THE ADDRESS PLUG FOR RPO5/6'S.)

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

013
 014
 015
 016
 017
 018
 019
 020
 021
 022
 023
 024
 025
 026
 027
 028
 029
 030
 031
 032
 033
 034
 035
 036
 037
 038
 039
 040
 041
 042
 043
 044
 045
 046
 047
 048
 049
 050
 051
 052
 053
 054
 055
 056
 057
 058
 059
 060
 061
 062
 063
 064
 065
 066
 067
 068

BITS FOR PORTS A & B ARE ASSERTED IN THE SAME BIT POSITION WHEN 'RPAS' (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 'RPDS' (DRIVE STATUS REGISTER) TO DETERMINE THE STATE OF THE SELECTED PORTS'S ATTENTION BIT.

6. ERRORS

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

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

7. MISCELLANEOUS

7.1 RESTRICTIONS

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 STACK POINTER

THE STACK IS INITIALLY SET TO 1100 AND EXTENDS DOWNWARD IN MEMORY.

369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424

7.5 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

- A. 'SCOPE' (IOT INSTRUCTION). THIS CALL IS PLACED BETWEEN EACH TEST IN THE INSTRUCTION. THIS ROUTINE ESTABLISHES THE ITERATION COUNT AND THE LOOP ON TEST AND LOOP ON ERROR ADDRESSES.
- B. 'ERROR' (EMT INSTRUCTION). THIS CALL IS USED TO REPORT ALL ERRORS. THE CALL IS FOLLOWED BY A NUMBER WHICH IDENTIFIES THE ERROR MESSAGE WHICH WILL BE TYPED.

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

- TYPE - TTY TYPEOUT ROUTINE
- TYPOC - TYPE OCTAL NUMBER (WITH LEADING ZERO)
- TYPOS - TYPE OCTAL NUMBER (NO LEADING ZEROS)
- TYPON - TYPE OCTAL NUMBER PER LAST CALL
- TYPDS - TYPE DECIMAL NUMBER WITH SIGN
- RDCHR - READ CHARACTER FROM TTY KEYBOARD
- RDLIN - READ A LINE FROM THE TTY KEYBOARD.
- RDOCT - READ AN OCTAL NUMBER FROM THE TTY KEYBOARD
- SAVREG - ROUTINE TO SAVE RO-R5
- RESREG - ROUTINE TO RESTORE RO-R5

7.6 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.
- B. TEST 4 AND TEST 5. THESE TESTS SET 'VV-A' AND 'VV-B' RESPECTIVLY. THESE TESTS MUST BE PERFORMED AT LEASE ONCE BEFORE TESTS 6 - 46 ARE RUN.

7.7 DISK SURFACE USAGE

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

7.8 TEST ITERATIONS

EACH TEST IS PERFORMED ONCE ON THE FIRST PASS THROUGH THE PROGRAM. ON THE SECOND AND SUBSEQUENT PASSES THROUGH THE PROGRAM, EACH TEST IS PERFORMED THE FOLLOWING NUMBER OF TIMES:

ITERATION COUNT

481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536

WAS DETECTED.

7.10 SPECIAL M7775 'DP' BOARD TESTS

THE PROGRAM CONTAINS 2 SPECIAL TESTS FOR THE M7775 'DP' BOARD TO VERIFY THE PROPER OPERATION OF THE PORT TIMEOUT ONE-SHOT. THESE TESTS ARE NOT RUN AS PART OF THE NORMAL SEQUENCE AND MUST BE SELECTED BY THE OPERATOR. THE TESTS ARE TEST 45 AND TEST 46.

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 (RPDS1) 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 RPDS1, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DESCREPNACY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

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 (RPDS1) THROUGH THE SEIZING PORT AND VERIFING THAT CORRECT STATUS IS SEEN. WHEN RPDS1 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 TEST 1 - DRIVE ACCESS TEST

VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS

A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RPD4/5/6, THAT THE DRIVE IS ONLINE (RPDS1 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.

8.4 TEST 2 - PORT 'A' SEIZE/TIMEOUT TEST

VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES THE DRIVE.

A. WRITE 0'S INTO RPDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.

537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592

- B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'B';
VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
- C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND 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.

8.5 TEST 3 - PORT 'B' SEIZE/TIMEOUT TEST

VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
THE DRIVE.

- A. WRITE 0'S INTO RPDS1 THROUGH PORT 'B'; VERIFY THAT THE DRIVE
HAS BEEN SEIZED.
- B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'A';
VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
- C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND 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.

8.6 TEST 4 - PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'

VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
FOR THE PORT UNDER TEST.

- A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE
DRIVE WAS SEIZED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
- B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE
'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
- C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.

8.7 TEST 5 - PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
FOR THE PORT UNDER TEST.

- A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'. VERIFY THAT THE
DRIVE WAS SEIZED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
- B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'. VERIFY THAT THE
'VV' BIT FOR PORT 'B' WAS SET.
- C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.

8.8 TEST 6 - TEST RELEASE, DRIVE SEIZED BY PORT 'A'

593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648

- TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- 8.9 TEST 7 - TEST RELEASE, DRIVE SEIZED BY PORT 'B'
- TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- 8.10 TEST 10 - TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
- 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.
- 8.11 TEST 11 - TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
- 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.
- 8.12 TEST 12 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
- 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 RPDS1 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 RH11 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.
- 8.13 TEST 13 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
- 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 RPDS1 THROUGH PORT 'B'.

649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704

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 RH11 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.

8.14 TEST 14 - TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

VERIFY THAT A MASSBUS INITIALIZE 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 RPDS1.
- C. ISSUE A MASSBUS CLEAR.
- 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.

8.15 TEST 15 - TEST RESET ATTENTION 'B' BY MASSBUS CLEAR

VERIFY THAT A MASSBUS INITIALIZE 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 RPDS1.
- C. ISSUE A MASSBUS CLEAR.
- 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.

8.16 TEST 16 - TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL

VERIFY THAT MASSBUS CLEAR DOES NOT RESET 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 NEITHER ATTENTION BIT HAS RESET.

8.17 TEST 17 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'

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

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.

- A. READ THE CONTROL REGISTER (RPCS1) 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.

8.18 TEST 20 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.

- A. READ THE CONTROL REGISTER (RPCS1) 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.

8.19 TEST 21 - TEST 'PORT REQUEST' FROM PORT 'A'

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 RPDS1.
- B. WRITE 0'S INTO RPDS1 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.

8.20 TEST 22 - TEST PORT REQUEST FROM PORT 'B'

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 RPDS1.
- B. WRITE 0'S INTO RPDS1 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.

8.21 TEST 23 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT

761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816

REQUEST'.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1. 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.

8.22 TEST 24 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT REQUEST'.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1. 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.

8.23 TEST 25 - TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'

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 RPDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
- 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.

8.24 TEST 26 - TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'

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 RPDS1.
- 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'.

817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872

- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.25 TEST 27 - TEST SEIZE BY WRITING ATTENTION BIT
TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER (RPAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER PORT.
 - A. WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; 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.
- 8.26 TEST 30 - TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
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.
- 8.27 TEST 31 - 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 RPDS1.
 - B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
 - C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
- 8.28 TEST 32 - 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 RPDS1.
 - B. WRITE 1'S INTO RPER1 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.
- 8.29 TEST 33 - TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
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 RPDS1.

873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928

- B. WRITE 1'S INTO RPER1 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 RPER1 HAS NOT BEEN CLEARED.
- D. CLEAR RPER1 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.

8.30 TEST 34 - TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET

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 RPDS1.
- B. WRITE 1'S INTO RPER1 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 RPER1 HAS NOT BEEN CLEARED.
- D. CLEAR RPER1 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.

8.31 TEST 35 - TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

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

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 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.

8.33 TEST 37 - TEST PORT 'A' ATTENTION AFTER A COMMAND

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

- A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
- B. WAIT FOR THE RECALIBRATE 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

929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984

TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

- 8.34 TEST 40 - TEST PORT 'B' ATTENTION AFTER A COMMAND
- TEST THE OPERATION OF THE PROT A AND PORT B ATTENTION BITS AFTER A COMMAND.
- ISSUE A RECALIBRATE COMMAND THROUGH PORT '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.
 - RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.35 TEST 41 - TEST PORT INTERACTION FROM PORT 'A'
- VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 - CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 - 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.
 - ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.36 TEST 42 - TEST PORT INTERACTION FROM PORT 'B'
- VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 - READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 - CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 - RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS

985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

- 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.
- 8.37 TEST 43 - TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
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 RPDS1.
C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.
- 8.38 TEST 44 - TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
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 RPDS1.
C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.
- 8.39 TEST 45 - TEST NO TIMEOUT THROUGH PORT 'A'
VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES. THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST BE SELECTED SEPARATELY.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 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 NOT BEEN RELEASED.
E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.40 TEST 46 - TEST NO TIMEOUT THROUGH PORT 'B'
VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES. THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST

1041
1042
1043
1044
1045
1046
1047
1048
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

BE SELECTED SEPARATELY.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 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 NOT BEEN RELEASED.
- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

9. PROGRAM LISTING

2

```
.TITLE MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1
;*COPYRIGHT (C) 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY C. HESS
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
;*
```

```
.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
```

001100

000011
000012

1097	000015	CR= 15	:: CODE FOR CARRIAGE RETURN
1098	000200	CRLF= 200	:: CODE FOR CARRIAGE RETURN-LINE FEED
1099	177776	PS= 177776	:: PROCESSOR STATUS WORD
1100		.EQUIV PS,PSW	
1101	177774	STKLMT= 177774	:: STACK LIMIT REGISTER
1102	177772	PIRQ= 177772	:: PROGRAM INTERRUPT REQUEST REGISTER
1103	177570	DSWR= 177570	:: HARDWARE SWITCH REGISTER
1104	177570	DDISP= 177570	:: HARDWARE DISPLAY REGISTER

*GENERAL PURPOSE REGISTER DEFINITIONS

1106		R0= %0	:: GENERAL REGISTER
1107	000000	R1= %1	:: GENERAL REGISTER
1108	000001	R2= %2	:: GENERAL REGISTER
1109	000002	R3= %3	:: GENERAL REGISTER
1110	000003	R4= %4	:: GENERAL REGISTER
1111	000004	R5= %5	:: GENERAL REGISTER
1112	000005	R6= %6	:: GENERAL REGISTER
1113	000006	R7= %7	:: GENERAL REGISTER
1114	000007	SP= %6	:: STACK POINTER
1115	000006	PC= %7	:: PROGRAM COUNTER

*PRIORITY LEVEL DEFINITIONS

1118		PR0= 0	:: PRIORITY LEVEL 0
1119	000000	PR1= 40	:: PRIORITY LEVEL 1
1120	000040	PR2= 100	:: PRIORITY LEVEL 2
1121	000100	PR3= 140	:: PRIORITY LEVEL 3
1122	000140	PR4= 200	:: PRIORITY LEVEL 4
1123	000200	PR5= 240	:: PRIORITY LEVEL 5
1124	000240	PR6= 300	:: PRIORITY LEVEL 6
1125	000300	PR7= 340	:: PRIORITY LEVEL 7
1126	000340		

*"SWITCH REGISTER" SWITCH DEFINITIONS

1127		SW15= 100000	
1128		SW14= 40000	
1129	100000	SW13= 20000	
1130	040000	SW12= 10000	
1131	020000	SW11= 4000	
1132	010000	SW10= 2000	
1133	004000	SW09= 1000	
1134	002000	SW08= 400	
1135	001000	SW07= 200	
1136	000400	SW06= 100	
1137	000200	SW05= 40	
1138	000100	SW04= 20	
1139	000040	SW03= 10	
1140	000020	SW02= 4	
1141	000010	SW01= 2	
1142	000004	SW00= 1	
1143	000002	.EQUIV SW09,SW9	
1144	000001	.EQUIV SW08,SW8	
1145		.EQUIV SW07,SW7	
1146		.EQUIV SW06,SW6	
1147		.EQUIV SW05,SW5	
1148		.EQUIV SW04,SW4	
1149		.EQUIV SW03,SW3	
1150		.EQUIV SW02,SW2	
1151			
1152			

```

1153      .EQUIV SW01,SW1
1154      .EQUIV SW00,SW0
1155
1156      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1157      100000      BIT15= 100000
1158      040000      BIT14= 40000
1159      020000      BIT13= 20000
1160      010000      BIT12= 10000
1161      004000      BIT11= 4000
1162      002000      BIT10= 2000
1163      001000      BIT09= 1000
1164      000400      BIT08= 400
1165      000200      BIT07= 200
1166      000100      BIT06= 100
1167      000040      BIT05= 40
1168      000020      BIT04= 20
1169      000010      BIT03= 10
1170      000004      BIT02= 4
1171      000002      BIT01= 2
1172      000001      BIT00= 1
1173      .EQUIV BIT09,BIT9
1174      .EQUIV BIT08,BIT8
1175      .EQUIV BIT07,BIT7
1176      .EQUIV BIT06,BIT6
1177      .EQUIV BIT05,BIT5
1178      .EQUIV BIT04,BIT4
1179      .EQUIV BIT03,BIT3
1180      .EQUIV BIT02,BIT2
1181      .EQUIV BIT01,BIT1
1182      .EQUIV BIT00,BIT0
1183
1184      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1185      000004      ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
1186      000010      RESVEC= 10         ;; RESERVED AND ILLEGAL INSTRUCTIONS
1187      000014      TBITVEC=14         ;; "T" BIT
1188      000014      TRTVEC= 14         ;; TRACE TRAP
1189      000014      BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
1190      000020      IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
1191      000024      PWRVEC= 24         ;; POWER FAIL
1192      000030      EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
1193      000034      TRAPVEC=34         ;; "TRAP" TRAP
1194      000060      TKVEC= 60          ;; TTY KEYBOARD VECTOR
1195      000064      TPVEC= 64          ;; TTY PRINTER VECTOR
1196      000240      PIRGVEC=240       ;; PROGRAM INTERRUPT REQUEST VECTOR
1197
1198      ;;*****
1199
1200      .SBTTL RH11 REGISTERS
1201
1202      ;;*****
1203
1204      ;CONTROL AND STATUS REGISTER 1 (RPCS1)
1205
1206      000100      IE= 100             ; INTERRUPT ENABLE (BIT #6)
1207      000200      RDY= 200           ; READY (BIT #7)
1208      000400      A16= 400           ; HIGH ORDER BUS ADDRESS BIT (BIT #8)

```



```

1209      001000      A17=      1000      ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
1210      002000      PSEL=     2000      ;PORT SELECT (BIT #10)
1211      020000      MCPE=    20000      ;MASSBUS PARITY ERROR (BIT #13)
1212      040000      TRE=     40000      ;TRANSFER ERROR (BIT #14)
1213      100000      SC=     100000     ;SPECIAL CONDITION (BIT #15)
1214
1215      ;WORD COUNT REGISTER (RPWC)
1216      ;(EACH BIT IS CALLED BY BIT NUMBER)
1217
1218      ;BUS ADDRESS REGISTER (RPBA)
1219      ;(EACH BIT IS CALLED BY BIT NUMBER)
1220
1221      ;CONTROL AND STATUS REGISTER 2 (RPCS2)
1222
1223      000001      US1=       1      ;UNIT SELECT (BIT #0)
1224      000002      US2=       2      ;UNIT SELECT (BIT #1)
1225      000004      US4=       4      ;UNIT SELECT (BIT #2)
1226      000010      BAI=      10      ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
1227      000020      PAT=      20      ;MASSBUS PARITY TEST (BIT #4)
1228      000040      CLR=      40      ;CLEAR (BIT #5)
1229      000100      IR=      100      ;INPUT READY (BIT #6)
1230      000200      OR=      200      ;OUTPUT READY (BIT #7)
1231      000400      MPE=     400      ;MASS BUS PARITY ERROR (BIT #8)
1232      001000      MXF=    1000      ;MISSED TRANSFER ERROR (BIT #9)
1233      002000      PGE=    2000      ;PROGRAM ERROR (BIT #10)
1234      004000      NEM=    4000      ;NON EXISTENT MEMORY (BIT #11)
1235      010000      NED=   10000      ;NON EXISTENT DRIVE (BIT #12)
1236      020000      UPE=   20000      ;UNIBUS PARITY ERROR (BIT #13)
1237      040000      WCE=   40000      ;WRITE CHECK ERROR (BIT #14)
1238      100000      DLT=  100000     ;DATA LATE (BIT #15)
1239
1240      ;DATA BUFFER REGISTER (RPDB)
1241      ;(EACH BIT IS CALLED BY BIT NUMBER)
1242
1243
1244      ;*****
1245
1246      .SBTTL  RP04/5/6 REGISTERS
1247
1248      ;*****
1249
1250      ;CONTROL AND STATUS 1 REGISTER. (#00)
1251
1252      000001      G0=       1      ;GO BIT (BIT #0)
1253      000002      F1=       2      ;FUNCTION CODE BIT #1
1254      000004      F2=       4      ;FUNCTION CODE BIT #2
1255      000010      F3=      10      ;FUNCTION CODE BIT #3
1256      000020      F4=      20      ;FUNCTION CODE BIT #4
1257      000040      F5=      40      ;FUNCTION CODE BIT #5
1258      004000      DVA=     4000      ;DEVICE AVAILABLE (BIT #11)
1259
1260      ;DRIVE STATUS REGISTER (RPDS1) (#01)
1261
1262      ;DFS=      1      DRIVE FORWARD 5"/SEC. (BIT #0)
1263      000002      DFF20=  2      ;DRIVE FORWARD 20"/SEC. (BIT #1)
1264      000004      DIGB=  4      ;DRIVE TO INNER GUARD BAND (BIT #2)

```

10-1

1265	000010	GRV=	10	;GO REVERSE (BIT #3)
1266	000020	DL64=	20	;DIFFERENCE LESS THAN 64 (BIT #4)
1267	000040	DE1=	40	;DIFFERENCE EQUALS 1 (BIT #5)
1268	000100	VV=	100	;VOLUME VALID (BIT #6)
1269	000200	DRY=	200	;DRIVE READY (BIT #7)
1270	000400	DFR=	400	;DRIVE PRESENT (BIT #8)
1271	001000	PGM=	1000	;PROGRAMABLE (BIT #9)
1272	002000	LST=	2000	;LAST SECTOR TRANSFERRED (BIT #10)
1273	004000	WRL=	4000	;WRITE LOCK (BIT #11)
1274	010000	MOL=	10000	;MEDIUM ON-LINE (BIT #12)
1275	020000	PIP=	20000	;POSITIONING OPERATION IN PROGRESS (BIT #13)
1276	040000	ERR=	40000	;COMPOSITE ERROR (BIT #14)
1277	100000	ATA=	100000	;ATTENTION ACTIVE (BIT #15)

;ERROR REGISTER #01 (RPER1) (#02)

1281	000001	ILF=	1	;ILLEGAL FUNCTION (BIT #0)
1282	000002	ILR=	2	;ILLEGAL REGISTER (BIT #1)
1283	000004	RMR=	4	;REGISTER MODIFICATION REFUSED (BIT #2)
1284	000010	PAR=	10	;PARITY ERROR (BIT #3)
1285	000020	FER=	20	;FORMAT ERROR (BIT #4)
1286	000040	WCF=	40	;WRITE CLOCK FAIL (BIT #5)
1287	000100	ECH=	100	;ECC HARD ERROR (BIT #6)
1288	000200	HCE=	200	;HEADER COMPARE ERROR (BIT #7)
1289	000400	HCRC=	400	;HEADER CRC ERROR (BIT #8)
1290	001000	AOE=	1000	;ADDRESS OVERFLOW ERROR (BIT #9)
1291	002000	IAE=	2000	;INVALID ADDRESS ERROR (BIT #10)
1292	004000	WLE=	4000	;WRITE LOCK ERROR (BIT #11)
1293	010000	DTE=	10000	;DRIVE TIMING ERROR (BIT #12)
1294	020000	OPI=	20000	;OPERATION INCOMPLETE (BIT #13)
1295	040000	UNS=	40000	;DRIVE UNSAFE (BIT #14)
1296	100000	DCK=	100000	;DATA CHECK ERROR (BIT 15)

;MAINTAINABILITY REGISTER (RPMR) (#03)

1300	000001	DMD=	1	;DIAGINOSTIC MODE (BIT #0)
1301	000002	MCLK=	2	;MAINTAINABILITY CLOCK (BIT #1)
1302	000004	MINX=	4	;MAINTAINABILITY INDEX (BIT #2)
1303	000010	MSTCK=	10	;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1304	000020	MRD=	20	;MAINTAINABILITY READ (BIT #4)
1305	000040	MWR=	40	;MAINTAINABILITY WRITE (BIT #5)
1306	000200	DTSY=	200	;MAINTAINABILITY SYNC DETECTED (BIT #7)

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

1310	000001	AT0=	1	;DEVICE 0 (BIT #0)
1311	000002	AT1=	2	;DEVICE 1 (BIT #1)
1312	000004	AT2=	4	;DEVICE 2 (BIT #2)
1313	000010	AT3=	10	;DEVICE 3 (BIT #3)
1314	000020	AT4=	20	;DEVICE 4 (BIT #4)
1315	000040	AT5=	40	;DEVICE 5 (BIT #5)
1316	000100	AT6=	100	;DEVICE 6 (BIT #6)
1317	000200	AT7=	200	;DEVICE 7 (BIT #7)

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

1318
1319
1320

```

1321
1322
1323
1324      000001      DT00= 1      ;DRIVE TYPE REGISTER (RPDT) (#06)
1325      000002      DT01= 2      ;DRIVE TYPE NUMBER BIT 1
1326      000004      DT02= 4      ;DRIVE TYPE NUMBER BIT 2
1327      000010      DT03= 10     ;DRIVE TYPE NUMBER BIT 3
1328      000020      DT04= 20     ;DRIVE TYPE NUMBER BIT 4
1329      000040      DT05= 40     ;DRIVE TYPE NUMBER BIT 5
1330      000100      DT06= 100    ;DRIVE TYPE NUMBER BIT 6
1331      000200      DT07= 200    ;DRIVE TYPE NUMBER BIT 7
1332      000400      DT08= 400    ;DRIVE TYPE NUMBER BIT 8
1333      004000      DRQ= 4000   ;DRIVE TYPE NUMBER BIT 9
1334      020000      MOH= 20000  ;DRIVE REQUEST REQUIRED (BIT #11)
1335      040000      TAP= 40000  ;MOVING HEAD (BIT #13)
1336      100000      NBA= 100000 ;TAPE DRIVE (BIT #14)
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376

```

```

;LOOK-AHEAD REGISTER (RPLA) (#07)
EXT1= 1      ;EXTENSION 1 (BIT #0)
EXT2= 2      ;EXTENSION 2 (BIT #1)
EXT4= 4      ;EXTENSION 3 (BIT #2)
EXT10= 10    ;EXTENSION 4 (BIT #3)
EXT20= 20    ;EXTENSION 5 (BIT #4)
EXT40= 40    ;EXTENSION 6 (BIT #5)
SC1= 100     ;SECTOR COUNT FIELD 0 (BIT #6)
SC2= 200     ;SECTOR COUNT FIELD 1 (BIT #7)
SC4= 400     ;SECTOR COUNT FIELD 2 (BIT #8)
SC10= 1000  ;SECTOR COUNT FIELD 3 (BIT #9)
SC20= 2000  ;SECTOR COUNT FIELD 4 (BIT #10)
TRK1= 4000   ;TRACK FIELD 1 (BIT #11)
TRK2= 10000  ;TRACK FIELD 2 (BIT #12)
TRK4= 20000  ;TRACK FIELD 3 (BIT #13)
TRK10= 40000 ;TRACK FIELD 4 (BIT #14)
TRK20= 100000 ;TRACK FIELD 5 (BIT #15)

;RPO4 ERROR REGISTER #2 (RPER2) (#10)
WCU= 1      ;WRITE CURRENT UNSAFE (BIT #0)
CSF= 2      ;CURRENT SINK FAILURE (BIT #1)
WSU= 4      ;WRITE SELECT UNSAFE (BIT #2)
CSU= 10     ;CURRENT SWITCH UNSAFE (BIT #3)
MSE= 20     ;MOTOR SEQUENCE ERROR (BIT #4)
TDF= 40     ;TRANSITIONS DETECTOR FAILURE (BIT #5)
TUF= 100    ;TRANSITIONS UNSAFE (BIT #6)
FEN= 200    ;FAILSAFE ENABLED (BIT #7)
WRU= 400    ;WRITE READY UNSAFE (BIT #8)
MHS= 1000   ;MULTIPLE HEAD SELECT (BIT #9)
NHS= 2000   ;NO HEAD SELECTION (BIT #10)
IXE= 4000   ;INDEX ERROR (BIT #11)
VU30= 10000 ;30VOLT UNSAFE (BIT #12)
PLU= 20000  ;PLO UNSAFE (BIT #13)
ACU= 100000 ;AC UNSAFE (BIT #15)

;RPO5/6 ERROR REGISTER #02 (RPER2) (#10)

```

1377	000001	WCU=	1	;WRITE CURRENT UNSAFE (BIT #0)
1378	000002	CSF=	2	;CURRENT SINK FAILURE (BIT #1)
1379	000004	WSU=	4	;WRITE SELECT UNSAFE (BIT #2)
1380	000010	CSU=	10	;CURRENT SWITCH UNSAFE (BIT #3)
1381	000020	RAW=	20	;READ AND WRITE (BIT #4)
1382	000040	TDF=	40	;TRANSITIONS DETECTOR FAILURE (BIT #5)
1383	000100	TUF=	100	;TRANSITIONS UNSAFE (BIT #6)
1384	000200	ABS=	200	;ABNORMAL STOP (BIT #7)
1385	000400	WRU=	400	;WRITE READY UNSAFE (BIT #8)
1386	001000	MHS=	1000	;MULTIPLE HEAD SELECT (BIT #9)
1387	002000	NHS=	2000	;NO HEAD SELECTION (BIT #10)
1388	004000	IXE=	4000	;INDEX ERROR (BIT #11)
1389	020000	PLU=	20000	;PLO UNSAFE (BIT #12)
1390				
1391		;OFFSET REGISTER (RPOF) (#11)		
1392				
1393	000001	OF25=	1	;OFFSET 25 MICRO INCHES (BIT #0)
1394	000002	OF50=	2	;OFFSET 50 MICRO INCHES (BIT #1)
1395	000004	OF100=	4	;OFFSET 100 MICRO INCHES (BIT #2)
1396	000010	OF200=	10	;OFFSET 200 MICRO INCHES (BIT #3)
1397	000020	OF400=	20	;OFFSET 400 MICRO INCHES (BIT #4)
1398	000040	OF800=	40	;OFFSET 800 MICRO INCHES (BIT #5)
1399	000200	OFREV=	200	;OFFSET NEGATIVE (REVERSE) (BIT #5)
1400	002000	HCI=	2000	;HEADER COMPARE INHIBIT (BIT #10)
1401	004000	ECI=	4000	;ERROR CORRECTION CODE INHIBIT (BIT #11)
1402	010000	FMT22=	10000	;FORMAT BIT (BIT #12)
1403				
1404		;DESIRED CYLINDER ADDRESS (RPCA) (#12)		
1405		;(EACH BIT IS CALLED BY BIT NUMBER)		
1406				
1407		;CURRENT CYLINDER ADDRESS (RPCC) (#13)		
1408		;(EACH BIT IS CALLED BY BIT NUMBER)		
1409				
1410		;SERIAL NUMBER REGISTER (RPSN) (#14)		
1411		;(EACH IS CALLED BY BIT NUMBER)		
1412				
1413		;RPO4 ERROR REGISTER #03 (RPER3) (#15)		
1414				
1415	000001	PSU=	1	;PACK SPEED UNSAFE (BIT #0)
1416	000002	VUF=	2	;VELOCITY UNSAFE (BIT #1)
1417	000010	UWR=	10	;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1418	000020	PRE=	20	;DISK PACK ROTATION ERROR (BIT #4)
1419	000040	ACL=	40	;AC LOW (BIT #5)
1420	000100	DCL=	100	;DC LOW (BIT #6)
1421	040000	SKI=	40000	;SEEK INCOMPLETE (BIT #14)
1422	100000	OCYL=	100000	;OFF CYLINDER (BIT #15)
1423				
1424		;RPO5/6 ERROR REGISTER #03 (RPER3) (#15)		
1425				
1426	000001	DCU=	1	;DC UNSAFE (BIT #0)
1427	000002	WAO=	2	;WRITE AND OFFSET (BIT #1)
1428	000040	ACL=	40	;AC LOW (BIT #5)
1429	000100	DCL=	100	;DC LOW (BIT #6)
1430	020000	OPE=	20000	;OPERATOR PLUG ERROR (BIT #13)
1431	040000	SKI=	40000	;SEEK INCOMPLETE (BIT #14)
1432	100000	OCYL=	100000	;OFF CYLINDER ERROR (BIT #15)

1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488

;ECC POSITION REGISTER (RPEC1) (#16)
;(EACH BIT IS CALLED BY BIT NUMBER)

;ECC PATTERN REGISTER (RPEC2) (#17)
;(EACH BIT IS CALLED BY BIT NUMBER)

;;*****

.SBTTL DEFINITIONS OF THE RH1!/RPO4/5/6 ADDRESS INDEXES

;;*****

000000	RPCS1=0	;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
000002	RPWC=2	;WORD COUNT REGISTER (NOT A DRIVE REG)
000004	RPBA=4	;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
000006	RPDA=6	;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
000010	RPCS2=10	;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
000012	RPDS1=12	;DRIVE STATUS REGISTER (DRIVE REG 01)
000014	RPER1=14	;ERROR REGISTER #1 (DRIVE REG. 02)
000016	RPAS=16	;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
000020	RPLA=20	;LOOK AHEAD REGISTER (DRIVE REG. 07)
000022	RPDB=22	;DATA BUFFER REGISTER (NOT A DRIVE REG.)
000024	RPMR=24	;MAINTAINABILITY REGISTER (DRIVE REG. 03)
000026	RPDT=26	;DRIVE TYPE REGISTER (DRIVE REG. 06)
000030	RPSN=30	;SERIAL NUMBER REGISTER (DRIVE REG. 10)
000032	RPOF=32	;OFFSET REGISTER (DRIVE REG. 11)
000034	RPCA=34	;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
000036	RPCC=36	;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
000040	RPER2=40	;ERROR REGISTER #2 (DRIVE REG. 14)
000042	RPER3=42	;ERROR REGISTER #3 (DRIVE REG. 15)
000044	RPEC1=44	;ECC POSITION REGISTER (DRIVE REG. 16)
000046	RPEC2=46	;ECC PATTERN REGISTER (DRIVE REG. 17)

.SBTTL TRAP CATCHER

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

000174	000000	. = 174	
DISPREG:	.WORD 0		::SOFTWARE DISPLAY REGISTER
SWREG:	.WORD 0		::SOFTWARE SWITCH REGISTER

.SBTTL ACT11 HOOKS

;;*****

		;HOOKS REQUIRED BY ACT11	
		SSVPC=.	;SAVE PC
		. = 46	
000046	056424	SENDAD	;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
		. = 52	
000052	020000	.WORD 20000	;2)SET LOC.52 TO 20000
		. = SSVPC	::RESTORE PC

C03

1489
1490
1491
1492
1493
1494
1495
1496
1497

```
.SBTTL STARTING ADDRESS = 200  
JMP START ;START THE PROGRAM  
.SBTTL START THE PROGRAM AND CHANGE THE RH11 ADDRESS = 204  
JMP START1 ;START AND CHANGE THE RH11 ADDRESS
```

.SBTTL COMMON TAGS

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

1498
1499
1500
1501
1502
1503
1504 001100
1505 001100
1506 001100 000000
1507 001102 000
1508 001103 000
1509 001104 000000
1510 001106 000000
1511 001110 000000
1512 001112 000000
1513 001114 000
1514 001115 001
1515 001116 000000
1516 001120 000000
1517 001122 000000
1518 001124 000000
1519 001126 000000
1520 001130 000000
1521 001132 000000
1522 001134 000
1523 001135 000
1524 001136 000000
1525 001140 177570
1526 001142 177570
1527 001144 177560
1528 001146 177562
1529 001150 177564
1530 001152 177566
1531 001154 000
1532 001155 002
1533 001156 012
1534 001157 000
1535 001160 000000
1536
1537 001162 000000
1538 001164 000000
1539 001166 000000
1540 001170 000000
1541 001172 000000
1542 001174 000000
1543 001176 000000
1544 001200 000000
1545 001202 177607 000377
1546 001206 077
1547 001207 015
1548 001210 000012
1549
1550 000015
1551 000012
1552 001212 172540
1553 001214 172542

.=1100

SCMTAG: .WORD 0
\$PASS: .WORD 0
\$STNM: .BYTE 00
\$ERFLG: .BYTE 00
\$ICNT: .WORD 00
\$LPADR: .WORD 00
\$LPERR: .WORD 00
\$ERTL: .WORD 00
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 00
\$BDADR: .WORD 00
\$GDDAT: .WORD 00
\$BDDAT: .WORD 00
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
\$DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$REGAD: .WORD 0
\$REGO: .WORD 0
\$TMP0: .WORD 0
\$TMP1: .WORD 0
\$TMP2: .WORD 0
\$TMP3: .WORD 0
\$TMP4: .WORD 0
\$TIMES: 0
\$ESCAPE: 0
\$BELL: .ASCIZ <207><377><377>
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCIZ <12>
CR = 15
LF = 12
\$LKCSR: .WORD 172540
\$LKCSB: .WORD 172542

:: START OF COMMON TAGS
:: CONTAINS PASS COUNT
:: CONTAINS THE TEST NUMBER
:: CONTAINS ERROR FLAG
:: CONTAINS SUBTEST ITERATION COUNT
:: CONTAINS SCOPE LOOP ADDRESS
:: CONTAINS SCOPE RETURN FOR ERRORS
:: CONTAINS TOTAL ERRORS DETECTED
:: CONTAINS ITEM CCYTR0L BYTE
:: CONTAINS MAX. ERRORS PER TEST
:: CONTAINS PC OF LAST ERROR INSTRUCTION
:: CONTAINS ADDRESS OF 'GOOD' DATA
:: CONTAINS ADDRESS OF 'BAD' DATA
:: CONTAINS 'GOOD' DATA
:: CONTAINS 'BAD' DATA
:: RESERVED--NOT TO BE USED
:: AUTOMATIC MODE INDICATOR
:: INTERRUPT MODE INDICATOR
:: ADDRESS OF SWITCH REGISTER
:: ADDRESS OF DISPLAY REGISTER
:: TTY KBD STATUS
:: TTY KBD BUFFER
:: TTY PRINTER STATUS REG. ADDRESS
:: TTY PRINTER BUFFER REG. ADDRESS
:: CONTAINS NULL CHARACTER FOR FILLS
:: CONTAINS # OF FILLER CHARACTERS REQUIRED
:: INSERT FILL CHARS. AFTER A "LINE FEED"
:: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
:: CONTAINS THE ADDRESS FROM
:: WHICH (\$REGO) WAS OBTAINED
:: CONTAINS ((\$REGAD)+0)
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: MAX. NUMBER OF ITERATIONS
:: ESCAPE ON ERROR ADDRESS
:: CODE FOR BELL
:: QUESTION MARK
:: CARRIAGE RETURN
:: LINE FEED
::*****

;; ADDR OF KW11-P STATUS REGISTER
;; ADDR OF KW11-P COUNTER BUFFER

1554	001216	000104	\$LPVEC: .WORD	104	; ADDR OF KW11-P VECTOR
1555	001220	177546	\$LKS: .WORD	177546	; ADDR OF KW11-L STATUS REGISTER
1556	001222	000100	\$LLVEC: .WORD	100	; ADDR OF KW11-L VECTOR
1557	001224	000000	PORTA: .WORD	0	; ADDRESS OF PORT A
1558	001226	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
1559	001230	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
1560	001232	000000	ASRI: .WORD	0	; ATA-A OR ATA-B = 1
1561	001234	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
1562	001236	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
1563	001240	000000	OPPR: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
1564	001242	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
1565	001244	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURRED
1566	001246	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
1567	001250	000000	RELERR: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
1568	001252	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
1569	001254	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
1570	001256	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
1571	001260	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
1572	001262	000000	TIMEAM: .WORD	0	; PORT A TIMEOUT VALUE - 25%
1573	001264	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
1574	001266	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
1575	001270	000000	TIMEBM: .WORD	0	; PORT B TIME VALUE - 25%
1576	001272	000000	TIMES: .WORD	0	; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
1577	001274	000000	KYBCTL: .WORD	0	; SINGLE TEST INDICATOR
1578	001276	000000	CHGADR: .WORD	0	; CHANGE THE RH11 ADDRESS INDICATOR
1579					
1580			;*****		
1581					
1582			.SBTTL RH11/RP04/5/6 UNIBUS AND VECTOR ADDRESSES		
1583					
1584			;*****		
1585					
1586	001300	176700	\$RPADR: .WORD	176700	; RH11/RP04/5/6 UNIBUS ADDRESS
1587	001302	000254	\$RPVEC: .WORD	254	; RH11 INTERRUPT VECTOR ADDRESS
1588					

1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

;ERROR 1

EM1	;WRONG DRIVE TYPE
DH1	
DT1	
DF1	

;ERROR 2

EM2	;DRIVE NOT ON LINE
DH1	
DT1	
DF1	

;ERROR 3

EM3	;SERIAL NUMBERS NOT THE SAME
DH3	
DT3	
DF1	

;ERROR 4

EM4	;DRIVE NOT SEIZED BY PORT 'N'
DH4	
DT7	
DF7	

;ERROR 5

EM5	;WRONG STATUS SEEN BY THE SEIZING PORT
DH5	
DT5	
DF5	

;ERROR 6

EM6	;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA
DH13	
DT13	

001304

001304 062506
 001306 066544
 001310 070302
 001312 070552

001314 062527
 001316 066544
 001320 070302
 001322 070552

001324 062551
 001326 066615
 001330 070316
 001332 070552

001334 062633
 001336 066664
 001340 070364
 001342 070565

001344 062664
 001346 067007
 001350 070332
 001352 070557

001354 062732
 001356 067257
 001360 070404

1645	001362	070557	DF5	
1646				
1647				;ERROR 7
1648				
1649	001364	063032	EM7	;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1650	001366	067063	DH7	
1651	001370	070364	DT7	
1652	001372	070565	DF7	
1653				
1654				;ERROR 10
1655				
1656	001374	063113	EM10	;REGISTER CONTENTS INCORRECT
1657	001376	067007	DH5	
1658	001400	070332	DT5	
1659	001402	070557	DF5	
1660				
1661				;ERROR 11
1662				
1663	001404	063143	EM11	;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1664	001406	067206	DH11	
1665	001410	070302	DT1	
1666	001412	070552	DF1	
1667				
1668				;ERROR 12
1669				
1670	001414	063227	EM12	;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1671	001416	067753	DH36	
1672	001420	070472	DT37	
1673	001422	070600	DF36	
1674				
1675				;ERROR 13
1676				
1677	001424	063277	EM13	; 'VOLUME VALID' BIT NOT SET BY READIN PRESET
1678	001426	067257	DH13	
1679	001430	070404	DT13	
1680	001432	070557	DF5	
1681				
1682				;ERROR 14
1683				
1684	001434	063364	EM14	; 'VOLUME VALID' SET ON THE OPPOSITE PORT
1685	001436	067257	DH13	
1686	001440	070404	DT13	
1687	001442	070557	DF5	
1688				
1689				;ERROR 15
1690				
1691	001444	063427	EM15	;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1692	001446	067063	DH7	
1693	001450	070364	DT7	
1694	001452	070565	DF7	
1695				
1696				;ERROR 16
1697				
1698	001454	063506	EM16	;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1699	001456	067063	DH7	
1700	001460	070364	DT7	

1701	001462	070565	DF7	
1702				
1703				;ERROR 17
1704				
1705	001464	063561	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
1706	001466	067063	DH7	
1707	001470	070364	DT7	
1708	001472	070565	DF7	
1709				
1710				;ERROR 20
1711				
1712	001474	063640	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
1713	001476	067753	DH36	
1714	001500	070472	DT37	
1715	001502	070600	DF36	
1716				
1717				;ERROR 21
1718				
1719	001504	063720	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
1720	001506	067753	DH36	
1721	001510	070472	DT37	
1722	001512	070600	DF36	
1723				
1724				;ERROR 22
1725				
1726	001514	063773	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
1727	001516	067377	DH22	
1728	001520	070422	DT22	
1729	001522	070574	DF31	
1730				
1731				;ERROR 23
1732				
1733	001524	064060	EM23	;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
1734	001526	067475	DH23	
1735	001530	070434	DT23	
1736	001532	070552	DF1	
1737				
1738				;ERROR 24
1739				
1740	001534	064126	EM24	;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1741	001536	067475	DH23	
1742	001540	070434	DT23	
1743	001542	070552	DF1	
1744				
1745				
1746				;ERROR 25
1747				
1748	001544	064205	EM25	;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1749	001546	067753	DH36	
1750	001550	070462	DT36	
1751	001552	070600	DF36	
1752				
1753				
1754				;ERROR 26
1755				
1756	001554	064250	EM26	;DRIVE NOT IN NEUTRAL AFTER RELEASE. REQUEST NOT SET

1757	001556	067377	DH22	
1758	001560	070422	DT22	
1759	001562	070574	DF31	
1760				
1761				;ERROR 27
1762				
1763	001564	064335	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1764	001566	067063	DH7	
1765	001570	070364	DT7	
1766	001572	070565	DF7	
1767				
1768				;ERROR 30
1769				
1770	001574	064413	EM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1771	001576	067753	DH36	
1772	001600	070462	DT36	
1773	001602	070600	DF36	
1774				
1775				;ERROR 31
1776				
1777	001604	064510	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SE
1778	001606	067654	DH31	
1779	001610	070450	DT31	
1780	001612	070574	DF31	
1781				
1782				;ERROR 32
1783				
1784	001614	064565	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1785	001616	067007	DH5	
1786	001620	070332	DT5	
1787	001622	070557	DF5	
1788				
1789				;ERROR 33
1790				
1791	001624	064636	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1792	001626	067753	DH36	
1793	001630	070462	DT36	
1794	001632	070600	DF36	
1795				
1796				;ERROR 34
1797				
1798	001634	064740	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1799	001636	067753	DH36	
1800	001640	070462	DT36	
1801	001642	070600	DF36	
1802				
1803				;ERROR 35
1804				
1805	001644	065043	EM35	;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
1806	001646	067753	DH36	
1807	001650	070472	DT37	
1808	001652	070600	DF36	
1809				
1810				;ERROR 36
1811				
1812	001654	065122	EM36	;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS

1813	001656	067753	DH36	
1814	001660	070462	DT36	
1815	001662	070600	DF36	
1816				
1817				;ERROR 37
1818				
1819	001664	065174	EM37	;DRIVE IS NON-EXISTENT
1820	001666	067753	DH36	
1821	001670	070472	DT37	
1822	001672	070600	DF36	
1823				
1824				;ERROR 40
1825				
1826	001674	065242	EM40	;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1827	001676	066544	DH1	
1828	001700	070434	DT23	
1829	001702	070552	DF1	
1830				
1831				;ERROR 41
1832				
1833	001704	065317	EM41	;TIMEOUT CLEARED ATTENTION BIT
1834	001706	067475	DH23	
1835	001710	070434	DT23	
1836	001712	070552	DF1	
1837				
1838				;ERROR 42
1839				
1840	001714	065361	EM42	;DRIVE NOT IN NEUTRAL OR SEIZED
1841	001716	070002	DH42	
1842	001720	070502	DT42	
1843	001722	070603	DF42	
1844				
1845				;ERROR 43
1846				
1847	001724	065447	EM43	;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1848	001726	070002	DH42	
1849	001730	070502	DT42	
1850	001732	070603	DF42	
1851				
1852				;ERROR 44
1853				
1854	001734	065524	EM44	;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1855	001736	070021	DH44	
1856	001740	070450	DT31	
1857	001742	070574	DF31	
1858				
1859				;ERROR 45
1860				
1861	001744	065601	EM45	;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1862	001746	066544	DH1	
1863	001750	070302	DT1	
1864	001752	070552	DF1	
1865				
1866				;ERROR 46
1867				
1868	001754	065660	EM46	;CAN'T ACCESS DRIVE THROUGH EITHER PORT

1869	001756	070117	DH46	
1870	001760	070510	DT46	
1871	001762	070574	DF31	
1872				
1873				;ERROR 47
1874				
1875	001764	065727	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
1876	001766	067475	DH23	
1877	001770	070434	DT23	
1878	001772	070552	DF1	
1879				
1880				;ERROR 50
1881				
1882	001774	066015	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
1883	001776	067257	DH13	
1884	002000	070404	DT13	
1885	002002	070557	DF5	
1886				
1887				;ERROR 51
1888				
1889	002004	066100	EM51	;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1890	002006	067007	DH5	
1891	002010	070332	DT5	
1892	002012	070557	DF5	
1893				
1894				;ERROR 52
1895				
1896	002014	066163	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1897	002016	067257	DH13	
1898	002020	070404	DT13	
1899	002022	070557	DF5	
1900				
1901				;ERROR 53
1902				;CAN'T READ ATTN BIT FROM OPPOSITE PORT
1903	002024	066261	EM53	
1904	002026	067475	DH23	
1905	002030	070302	DT1	
1906	002032	070552	DF1	
1907				
1908				;ERROR 54
1909				
1910	002034	066342	EM54	;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1911	002036	067377	DH22	
1912	002040	070522	DT54	
1913	002042	070574	DF31	
1914				
1915				;ERROR 55
1916				
1917	002044	066435	EM55	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1918	002046	070215	DH55	
1919	002050	070534	DT55	
1920	002052	070605	DF55	
1921				
1922				;ERROR 56
1923				
1924	002054	066502	EM56	;RH11 DIDN'T RESPOND TO ADDRESSING

```

1925 002056 070273          DH56
1926 002060 070546          DT56
1927 002062 070611          DF56
1928
1929
1930
1931 ;*****
1932 .SBTTL  STARTUP AND INITIALIZATION ROUTINES
1933 ;*****
1934
1935
1936
1937 002064 005037 001276  START:  CLR   CHGADR      ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
1938 002070 000403          BR     START2      ;GO TO THE START
1939 002072 012737 177777 001276  START1: MOV   #-1,CHGADR ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
1940 002100 000005          START2: RESET      ;CLEAR THE BUS
1941 .SBTTL  INITIALIZE THE COMMON TAGS
1942 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1943 002102 012706 001100  MOV   $CMTAG,R6    ;;FIRST LOCATION TO BE CLEARED
1944 002106 005026          CLR   (R6)+        ;;CLEAR MEMORY LOCATION
1945 002110 022706 001140  CMP   $SWR,R6 ;;DONE?
1946 002114 001374          BNE   -6           ;;LOOP BACK IF NO
1947 002116 012706 001100  MOV   $STACK,SP   ;;SETUP THE STACK POINTER
1948 ;;INITIALIZE A FEW VECTORS
1949 002122 012737 056670 000020  MOV   $$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1950 002130 012737 000340 000022  MOV   #340,@#IOTVEC+2 ;;LEVEL 7
1951 002136 012737 057122 000030  MOV   $ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1952 002144 012737 000340 000032  MOV   #340,@#EMTVEC+2 ;;LEVEL 7
1953 002152 012737 061722 000034  MOV   $TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1954 002160 012737 000340 000036  MOV   #340,@#TRAPVEC+2;LEVEL 7
1955 002166 013737 056270 056262  MOV   $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
1956 002174 005037 001176          CLR   $TIMES      ;;INITIALIZE NUMBER OF ITERATIONS
1957 002200 005037 001200          CLR   $ESCAPE     ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1958 002204 112737 000001 001115  MOVB  #1,$ERMAX   ;;ALLOW ONE ERROR PER TEST
1959 002212 012737 002212 001106  MOV   #,$SLPADR   ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1960 002220 012737 002220 001110  MOV   #,$SLPERR   ;;SETUP THE ERROR LOOP ADDRESS
1961 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1962 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1963 002226 013746 000004          MOV   @#ERRVEC,-(SP) ;;SAVE ERROR VECTOR
1964 002232 012737 002266 000004  MOV   #64,$@#ERRVEC ;;SET UP ERROR VECTOR
1965 002240 012737 177570 001140  MOV   #DSWR,$SWR  ;;SETUP FOR A HARDWARE SWICH REGISTER
1966 002246 012737 177570 001142  MOV   #DDISP,$DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1967 002254 022777 177777 176656  CMP   #-1,$SWR   ;;TRY TO REFERENCE HARDWARE SWR
1968 002262 001012          BNE   66$        ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1969 ;;AND THE HARDWARE SWR IS NOT = -1
1970 002264 000403          BR    65$        ;;BRANCH IF NO TIMEOUT
1971 002266 012716 002274 64$:  MOV   #65$,(SP)  ;;SET UP FOR TRAP RETURN
1972 002272 000002          RTI
1973 002274 012737 000176 001140 65$:  MOV   #SWREG,$SWR ;;POINT TO SOFTWARE SWR
1974 002302 012737 000174 001142  MOV   #DISPREG,$DISPLAY
1975 002310 012637 000004 66$:  MOV   (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR
1976
1977 002314 005227 177777          INC   #-1        ;FIRST START ?
1978 002320 001002          BNE   1$        ;BR IF NOT
1979 002322 104401 062010          TYPE  TITLE     ;TYPE PROGRAM NAME
1980 002326 004737 060332 1$:  JSR   PC,$TKINT ;SETUP THE TTY KEYBOARD

```

M03

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 38
 DZRJEA.CMB 02-NOV-76 18:40 GET VALUE FOR SOFTWARE SWITCH REGISTER

```

1981          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1982 002332 005737 000042          TST      0#42          ;;ARE WE RUNNING UNDER XXDP/ACT?
1983 002336 001006          BNE      67$          ;;BRANCH IF YES
1984 002340 023727 001140 000176    CMP      SWR,#SWREG    ;;SOFTWARE SWITCH REG SELECTED?
1985 002346 001005          BNE      68$          ;;BRANCH IF NO
1986 002350 104406          GTSWR                    ;;GET SOFT-SWR SETTINGS
1987 002352 000403          BR       68$
1988 002354 112737 000001 001134 67$:  MOVB     #1,$AUTOB    ;;SET AUTO-MODE INDICATOR
1989 002362          E8$:
1990 002362 004737 002754          JSR     PC,CHANGE     ;;CHECK/CHANGE THE RH11 ADDRESS
1991 002366 104401 062117          TYPE   ,ENTERA      ;;ENTER DRIVE ADDRESS
1992 002372 104412          RDOCT                    ;;GET THE ADDRESS
1993 002374 012637 001224          MOV     (SP)+,PORTA   ;;STORE THE ADDRESS
1994 002400 023727 001224 000007    CMP     PORTA,#7      ;;SEE IF ADDRESS TOO LARGE
1995 002406 101403          BLOS   2$            ;;BR IF NOT
1996 002410 104401 062147          TYPE   ,ADRERR      ;;TYPE ADDRESS ERROR MESSAGE
1997 002414 000744          BR     1$            ;;TRY AGAIN
1998 002416 013737 001224 001226 2$:  MOV     PORTA,PORTB   ;;GENERATE THE PORT B ADDRESS
1999 002424 005237 001226          INC     PORTB        ;;INCREMENT THE ADDRESS
2000 002430 042737 000016 001226    BIC     #16,PORTB    ;;LEAVE BIT 0
2001 002436 013746 001224          MOV     PORTA,-(SP)   ;;PUT PORT A ADDRESS ON THE STACK
2002 002442 042716 177771          BIC     #1C6,(SP)    ;;SAVE BITS 1 & 2
2003 002446 052637 001226          BIS     (SP)+,PORTB  ;;SET BITS 1 & 2 IN PORT B ADDRESS
2004 002452 104401 062171          TYPE   ,PORTAIS     ;;'PORT A ADDRESS IS '
2005 002456 013746 001224          MOV     PORTA,-(SP)  ;;SAVE PORTA FOR TYPEOUT
2006          ;;TYPE PORT A ADDRESS
2007          GO TYPE--OCTAL ASCII
2008 002462 104403          TYPOS                    ;;TYPE 1 DIGIT(S)
2009 002464 001          .BYTE  1
2010 002465 000          .BYTE  0
2011 002466 104401 062217          TYPE   ,PORTBIS    ;;SUPPRESS LEADING ZEROS
2012 002472 013746 001226          MOV     PORTB,-(SP) ;;'PORT B ADDRESS IS '
2013          ;;SAVE PORTB FOR TYPEOUT
2014          ;;TYPE PORT B ADDRESS
2015          GO TYPE--OCTAL ASCII
2016 002476 104403          TYPOS                    ;;TYPE 1 DIGIT(S)
2017 002500 001          .BYTE  1
2018 002501 000          .BYTE  0
2019          ;;SUPPRESS LEADING ZEROS
2020 002502 104401 001207          TYPE   ,$CRLF      ;;ANOTHER CR-LF
2021 002506 013737 001224 001230    MOV     PORTA,PORTC  ;;GENERATE ADDRESS OF DRIVE NOT TESTED
2022 002514 062737 000006 001230    ADD     #6,PORTC     ;;COMPLEMENT SOME BITS
2023 002522 042737 177770 001230    BIC     #1C7,PORTC  ;;SAVE ONLY LOWER BITS
2024 002530 013701 001224          MOV     PORTA,R1     ;;USE PORT A ADDRESS AS INDEX
2025 002534 116137 070726 001232    MOVB   ATABIT(R1),ASR1 ;;GET ATTENTION BIT FOR DRIVE
2026 002542 005037 001256          CLR     TIMEA        ;;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2027 002546 005037 001260          CLR     TIMEAP       ;;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2028 002552 005037 001264          CLR     TIMEB        ;;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2029 002556 005037 001266          CLR     TIMEBP       ;;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2030 002562 004737 056444          JSR     PC,CKCLK     ;;SETUP CLOCK
2031 002566 000137 002602          JMP     EXEC         ;;CLOCK HAS BEEN STARTED
2032 002572 104401 062245          TYPE   ,NOCLOCK     ;;NO CLOCK ON SYSTEM
2033 002576 000000          3$:  HALT                    ;;FATAL ERROR
2034 002600 000776          BR     3$           ;;INTERLOCK THE HALT
2035          ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
2036  EXEC:  RESET                    ;;CLEAR EVERYTHING
          CLR     PS           ;;CLEAR THE PROCESSOR STATUS WORD
          TYPE   , $CRLF      ;;CR-LF

```



```

2037 002614 013700 001300      MOV      $RPADR,RO      ;RH11 ADDRESS FOR INDEXING
2038 002620 012706 001100      MOV      #STACK,SP     ;LOAD STACK POINTER
2039 002624 004737 056444      JSR      PC,CKCLK      ;START THE CLOCK
2040 002630 000240                NOP                    ;RETURN IF NO CLOCK
2041 002632 004737 060332      JSR      PC,$TKINT     ;INITIALIZE THE KEYBOARD
2042 002636 005037 001274      CLR      KYBCTL        ;CLEAR SINGLE TEST INDICATOR
2043 002642 005037 001100      CLR      $PASS        ;CLEAR THE PASS COUNT
2044 002646 112737 000001 001115  MOVB     #1,$ERMAX     ;SET ERROR MAX TO 1
2045 002654 012737 002654 001106  MOV      #,$LPADR     ;INITIAL SETTING FOR LOOP ADDRESS
2046 002662 012737 002662 001110  MOV      #,$LPERR     ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
2047 002670 104401 062311 1$:      TYPE    ,TESTNO      ;ASK FOR TEST NUMBER
2048 002674 104412                RDOCT                ;GET THE NUMBER
2049 002676 012601                MOV      (SP)+,R1     ;PUT ENTRY INTO R1
2050 002700 001002                BNE     2$           ;BR IF NOT ZERO
2051 002702 000137 003070                JMP     TST1        ;ENTER ZERO - PERFORM ALL TESTS
2052 002706 020137 070736 2$:      CMP     R1,MAXTN     ;SEE IF NUMBER GREATER THAN MAXIMUM
2053 002712 003403                BLE     3$           ;BR IF LESS OR EQUAL
2054 002714 104401 062334                TYPE    ,BADNO      ;BAD ENTRY
2055 002720 000763                BR      1$           ;TRY AGAIN
2056 002722 005301 3$:      DEC     R1           ;DECREMENT ENTRY
2057 002724 006301                ASL     R1           ;SHIFT IT LEFT
2058 002726 016137 070612 002752  MOV     TSTADR(R1),4$ ;GET THE TEST ADDRESS
2059 002734 005237 001274                INC     KYBCTL       ;SET SINGLE TEST INDICATOR
2060 002740 012737 000001 001104  MOV     #1,$ICNT     ;PRESET ITERATION COUNT
2061 002746 000177 000000                JMP     @4$          ;GO TO THE SELECTED TEST
2062 002752 000000 4$:      .WORD  0            ;TEST ADDRESS GOES HERE
2063
2064                ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
2065
2066 002754 005737 001276  CHANGE: TST     CHGADR    ;CHANGE THE ADDRESS ?
2067 002760 001421                BEQ     3$           ;BR IF NOT
2068 002762 005037 001276                CLR     CHGADR       ;CLEAR THE INDICATOR
2069 002766 104401 062374 1$:      TYPE    ,ADDRIS     ;TYPE OUT WHAT THE PRESENT ADDRESS IS
2070 002772 013746 001300                MOV     $RPADR,-(SP) ;PUT THE ADDRESS ON THE STACK
2071 002776 104402                TYPE    ,NTRH11     ;TYPE THE ACTUAL ADDRESS
2072 003000 104401 001207                TYPE    ,NTRH11     ;CR-LF
2073 003004 104401 062454                TYPE    ,NTRH11     ;ASK FOR NEW ADDRESS
2074 003010 104412                RDOCT
2075 003012 005716                TST     (SP)         ;O OR 'CR' ENTERED ?
2076 003014 001402                BEQ     2$           ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
2077 003016 011637 001300                MOV     (SP),$RPADR  ;NEW RH11 ADDRESS
2078 003022 005726 2$:      TST     (SP)+       ;CORRECT THE STACK POINTER
2079 003024 012737 003044 000004 3$:      MOV     #4$,@#4     ;LOAD TRAP ADDRESS
2080 003032 013700 001300                MOV     $RPADR,RO    ;RH11 ADDRESS
2081 003036 005760 000002                TST     RPWC(RO)    ;SEE IF RH11 RESPONDS AT THAT ADDRESS
2082 003042 000404                BR      5$           ;BR, RH11 ALIVE AT PRESENT ADDRESS
2083 003044 104056 4$:      ERROR   56         ;NO RESPONSE TO ADDRESS
2084 003046 062706 000004                ADD     #4,SP        ;RESET THE STACK POINTER
2085 003052 000745                BR      1$           ;GET ADDRESS AGAIN
2086 003054 012737 000006 000004 5$:      MOV     #6,@#4     ;RESTORE THE VECTOR
2087 003062 000207                RTS     PC           ;RETURN
2088
2089                ;*****
2090
2091                .SBTTL  *** TESTS ***
2092

```

2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148

003064 013700 001300

003070
003070 005737 001274
003074 001406
003076 100002
003100 000137 002602
003104 012737 177777 001274
003112 112737 000001 001102
003120 012737 003142 001106
003126 012737 003142 001110
003134 012737 000001 001176
003142 012706 001100

003146 113760 001224 000010
003154 013737 001224 001234
003162 005760 000010
003166 005037 001244
003172 016037 000000 001126
003200 012737 000010 001122
003206 060037 001122
003212 005037 001124
003216 013737 001126 001164
003224 042737 167777 001164
003232 023737 001124 001164
003240 001414
003242 013737 001126 001174
003250 042737 010000 001174
003256 053737 001174 001124
003264 104037
003266 005137 001244
003272 000240
003274 005737 001244
003300 001403
003302 012760 000040 000010
003310 113760 001226 000010
003316 013737 001226 001234

```
*****  
TST1AA: MOV SRPADR,RO ;RESTORE RC AFTER END OF PASS  
*****  
TEST 1 DRIVE ACCESS TEST  
*  
* VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS  
*  
* A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE  
* DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RPO5! 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: MOVB #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  
*****  
;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B  
MOVB PORTA,RPCS2(RO) ;SELECT PORT A  
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
TST RPO5!(RO) ;SEE IF DRIVE (PORT A) PRESENT  
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR  
MOV RPCS2(RO), $BDDAT ;GET CONTENTS OF RPCS2  
MOV #RPCS2,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE  
ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS  
CLR $GDDAT ;WHAT REGISTER SHOULD BE  
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'  
BIC #NED,$TMP0 ;SAVE SPECIFIED BITS  
CMP $GDDAT,$TMP0 ;COMPARE THE BITS  
BEQ 645 ;BR IF OK  
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'  
BIC #NED,$TMP4 ;CLEAR THE MASKED BITS  
BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT  
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,RPCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'  
MOVB PORTB,RPCS2(RO) ;SELECT PORT B  
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```

2149 003324 005760 000012          TST      RPS1(RO)          ;SEE IF DRIVE (PORT B) PRESENT
2150 003330 005037 001244          CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2151 003334 016037 000010 001126    MOV      RPCS2(RO), $BDDAT ;GET CONTENTS OF RPCS2
2152 003342 012737 000010 001122    MOV      #RPCS2, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2153 003350 060037 001122          ADD      RO, $B0ADR       ;ADD RH11 BASE ADDRESS
2154 003354 005037 001124          CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
2155 003360 013737 001126 001164    MOV      $BDDAT, $TMP0    ;MOVE REGISTER CONTENTS TO 'STMP0'
2156 003366 042737 167777 001164    BIC      #1CND, $TMP0     ;SAVE SPECIFIED BITS
2157 003374 023737 001124 001164    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
2158 003402 001414          BEQ      66$             ;BR IF OK
2159 003404 013737 001126 001174    MOV      $BDDAT, $TMP4    ;COPY 'BAD DATA'
2160 003412 042737 010000 001174    BIC      #NED, $TMP4      ;CLEAR THE MASKED BITS
2161 003420 053737 001174 001124    BIS      $TMP4, $GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2162 003426 104037          ERROR    37              ;TYPE MESSAGE 37
2163 003430 005137 001244          COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2164 003434 000240          66$: NOP
2165 003436 005737 001244          TST      CKERR           ;WAS 'NED' SET ?
2166 003442 001403          BEQ      .+10            ;BR IF NOT
2167 003444 012760 000040 000010    MOV      #CLR, RPCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2168
2169
2170 ;*****
2171 ;CONFIRM THAT DRIVE IS AN RPO4/5/6 AND IS DUAL PORT
2172 003452 113760 001224 000010    MOV      PORTA, RPCS2(RO) ;SELECT PORT A
2173 003460 013737 001224 001234    MOV      PORTA, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2174 003466 005037 001244          CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2175 003472 016037 000026 001126    MOV      RPDT(RO), $BDDAT ;GET CONTENTS OF RPDT
2176 003500 012737 000026 001122    MOV      #RPDT, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2177 003506 060037 001122          ADD      RO, $B0ADR       ;ADD RH11 BASE ADDRESS
2178 003512 012737 024020 001124    MOV      #24020, $GDDAT  ;WHAT REGISTER SHOULD BE
2179 003520 013737 001126 001164    MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'STMP0'
2180 003526 042737 000003 001164    BIC      #1C177774, $TMP0 ;SAVE SPECIFIED BITS
2181 003534 023737 001124 001164    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
2182 003542 001414          BEQ      68$             ;BR IF OK
2183 003544 013737 001126 001174    MOV      $BDDAT, $TMP4    ;COPY 'BAD DATA'
2184 003552 042737 177774 001174    BIC      #177774, $TMP4   ;CLEAR THE MASKED BITS
2185 003560 053737 001174 001124    BIS      $TMP4, $GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2186 003566 104001          ERROR    1              ;TYPE MESSAGE 1
2187 003570 005137 001244          COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2188 003574 000240          68$: NOP
2189 003576 113760 001226 000010    MOV      PORTB, RPCS2(RO) ;SELECT PORT B
2190 003604 013737 001226 001234    MOV      PORTB, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2191 003612 005037 001244          CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2192 003616 016037 000026 001126    MOV      RPDT(RO), $BDDAT ;GET CONTENTS OF RPDT
2193 003624 012737 000026 001122    MOV      #RPDT, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2194 003632 060037 001122          ADD      RO, $B0ADR       ;ADD RH11 BASE ADDRESS
2195 003636 012737 024020 001124    MOV      #24020, $GDDAT  ;WHAT REGISTER SHOULD BE
2196 003644 013737 001126 001164    MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'STMP0'
2197 003652 042737 000003 001164    BIC      #1C177774, $TMP0 ;SAVE SPECIFIED BITS
2198 003660 023737 001124 001164    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
2199 003666 001414          BEQ      70$             ;BR IF OK
2200 003670 013737 001126 001174    MOV      $BDDAT, $TMP4    ;COPY 'BAD DATA'
2201 003676 042737 177774 001174    BIC      #177774, $TMP4   ;CLEAR THE MASKED BITS
2202 003704 053737 001174 001124    BIS      $TMP4, $GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2203 003712 104001          ERROR    1              ;TYPE MESSAGE 1
2204 003714 005137 001244          COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR

```

2205 003720 000240
2206
2207
2208
2209
2210 003722 113760 001224 000010
2211 003730 013737 001224 001234
2212 003736 005037 001244
2213 003742 016037 000012 001126
2214 003750 012737 000012 001122
2215 003756 060037 001122
2216 003762 012737 001000 001124
2217 003770 013737 001126 001164
2218 003776 042737 176777 001164
2219 004004 023737 001124 001164
2220 004012 001414
2221 004014 013737 001126 001174
2222 004022 042737 001000 001174
2223 004030 053737 001174 001124
2224 004036 104045
2225 004040 005137 001244
2226 004044 000240
2227 004046 005037 001244
2228 004052 016037 000012 001126
2229 004060 012737 000012 001122
2230 004066 060037 001122
2231 004072 012737 010600 001124
2232 004100 013737 001126 001164
2233 004106 042737 167177 001164
2234 004114 023737 001124 001164
2235 004122 001414
2236 004124 013737 001126 001174
2237 004132 042737 010600 001174
2238 004140 053737 001174 001124
2239 004146 104002
2240 004150 005137 001244
2241 004154 000240
2242 004156 113760 001226 000010
2243 004164 013737 001226 001234
2244 004172 005037 001244
2245 004176 016037 000012 001126
2246 004204 012737 000012 001122
2247 004212 060037 001122
2248 004216 012737 001000 001124
2249 004224 013737 001126 001164
2250 004232 042737 176777 001164
2251 004240 023737 001124 001164
2252 004246 001414
2253 004250 013737 001126 001174
2254 004256 042737 001000 001174
2255 004264 053737 001174 001124
2256 004272 104045
2257 004274 005137 001244
2258 004300 000240
2259 004302 005037 001244
2260 004306 016037 000012 001126

705: NOP

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

MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, \$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, \$BDAOR ;ADD RH11 BASE ADDRESS
MOV #PGM, \$GDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT, \$TMPD ;MOVE REGISTER CONTENTS TO '\$TMPD'
BIC #1CPGM, \$TMPD ;SAVE SPECIFIED BITS
CMP \$GDDAT, \$TMPD ;COMPARE THE BITS
BEQ 725 ;BR IF OK
MOV \$BDDAT, \$TMP4 ;COPY 'BAD DATA'
BIC #PGM, \$TMP4 ;CLEAR THE MASKED BITS
BIS \$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 45 ;TYPE MESSAGE 45
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

725: NOP

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, \$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, \$BDAOR ;ADD RH11 BASE ADDRESS
MOV #MOL!DPR!DRY, \$GDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT, \$TMPD ;MOVE REGISTER CONTENTS TO '\$TMPD'
BIC #1C10600, \$TMPD ;SAVE SPECIFIED BITS
CMP \$GDDAT, \$TMPD ;COMPARE THE BITS
BEQ 745 ;BR IF OK
MOV \$BDDAT, \$TMP4 ;COPY 'BAD DATA'
BIC #10600, \$TMP4 ;CLEAR THE MASKED BITS
BIS \$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 2 ;TYPE MESSAGE 2
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

745: NOP

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, \$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, \$BDAOR ;ADD RH11 BASE ADDRESS
MOV #PGM, \$GDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT, \$TMPD ;MOVE REGISTER CONTENTS TO '\$TMPD'
BIC #1CPGM, \$TMPD ;SAVE SPECIFIED BITS
CMP \$GDDAT, \$TMPD ;COMPARE THE BITS
BEQ 765 ;BR IF OK
MOV \$BDDAT, \$TMP4 ;COPY 'BAD DATA'
BIC #PGM, \$TMP4 ;CLEAR THE MASKED BITS
BIS \$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 45 ;TYPE MESSAGE 45
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

765: NOP

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1

```

2261 004314 012737 000012 001122      MOV      #RPS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2262 004322 060037 001122      ADD      R0,$BDADR ;ADD RH11 BASE ADDRESS
2263 004326 012737 010600 001124      MOV      #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
2264 004334 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
2265 004342 042737 167177 001164      BIC      #1C10600,$TMP0 ;SAVE SPECIFIED BITS
2266 004350 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
2267 004356 001414      BEQ      78$ ;BR IF OK
2268 004360 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
2269 004366 042737 010600 001174      BIC      #10600,$TMP4 ;CLEAR THE MASKED BITS
2270 004374 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2271 004402 104002      ERROR   2 ;TYPE MESSAGE 2
2272 004404 005137 001244      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2273 004410 000240      78$:    NOP

```

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

```

2278 004412 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
2279 004420 016037 000030 001124      MOV      RPSN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
2280 004426 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B
2281 004434 016037 000030 001126      MOV      RPSN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
2282 004442 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ARE THEY THE SAME ?
2283 004450 001406      BEQ      1$ ;BR IF THEY ARE
2284 004452 104003      ERROR   3 ;REPORT THE ERROR
2285 004454 032777 100000 174456      BIT      #SW15,$SWR ;HALT ON ERROR ?
2286 004462 001001      BNE      1$ ;BR IF SET - PROGRAM HAS ALREADY HALTED
2287 004464 000000      HALT ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
2288 004466 000004      1$:    SCOPE ;LOOP ?

```

*TEST 2 PORT 'A' SEIZE/TIMEOUT TEST

```

*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
*THE DRIVE.
*
*A. WRITE 0'S INTO RPS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE
*HAS BEEN SEIZED.
*
*B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'B';
*VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
*
*C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND 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.
*

```

*ST2:

```

2309 004470      TST      KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2310 004470 005737 001274      BEQ      2$ ;BR IF NOT
2311 004474 001406      BPL      1$ ;BR IF JUST ENTERED TEST
2312 004476 100002      JMP      EXEC ;RETURN & GET NEXT TEST NUMBER
2313 004500 000137 002602      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2314 004504 012737 177777 001274 1$:    MOVVB   #2,$STNM ;TEST NUMBER
2315 004512 112737 000002 001102 2$:    MOV      #TEST2,$LPAOR ;LOAD LOOP ON TEST ADDRESS
2316 004520 012737 004542 001106

```

F04

```

2317 004526 012737 001542 001110      MOV      #TEST2,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
2318 004534 012737 000012 001176      MOV      #10,$TIMES        ;DO 10. ITERATIONS
2319 004542 012706 001100      TEST2:  MOV      #STACK,SP    ;LOAD THE STACK POINTER
2320 004546 012737 000240 177776      MOV      #(<5*32.>),@#PS    ;SET PRIORITY TO 5 IN CASE LOOPING
2321 004554 005037 001256      CLR      TIMEA             ;CLEAR TIMEOUT VALUE FOR PORT A
2322 004560 005037 001260      CLR      TIMEAP           ;CLEAR UPPER TIMEOUT TOLERANCE
2323 004564 005037 001262      CLR      TIMEAM           ;CLEAR LOWER TIMEOUT TOLERANCE
2324
2325      ;:*****
2326      ;START THE TIMER
2327
2328 004570 005037 001252      CLR      TIME             ;CLEAR THE ELAPSED TIME COUNTER
2329 004574 012737 003720 001254      MOV      #2000.,WATCH    ;SET WATCH TO 2000 MS
2330
2331      ;:*****
2332
2333      ;SEIZE THE DRIVE THROUGH PORT A
2334
2335 004602 113760 001224 000010      MOVVB   PORTA,RPCS2(RO)   ;SELECT PORT A
2336 004610 013737 001224 001236      MOV     PORTA,SEIZPT     ;STORE SEIZING PORT'S ADDRESS
2337 004616 005060 000012      CLR     RPDS1(RO)        ;WRITE RPDS1
2338 004622 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)   ;SELECT PORT B
2339 004630 013737 001226 001234      MOV     PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2340 004636 013737 001226 001240      MOV     PORTB,OPPR       ;'OPPOSITE' PORT ADDRESS
2341 004644 016037 000012 001126      MOV     RPDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
2342 004652 010037 001122      MOV     RO,$BDAOR        ;R#11 BASE ADDRESS
2343 004656 062737 000012 001122      ADD     #RPDS1,$BDAOR    ;GENERATE BAD REGISTER ADDRESS
2344 004664 005037 001124      CLR     $GDDAT           ;REGISTER SHOULD BE ZERO
2345 004670 023737 001124 001126      CMP     $GDDAT,$BDDAT    ;IS THE REGISTER ZERO
2346 004676 001403      BEQ     64$              ;BR IF IT IS
2347 004700 104004      ERROR  4                 ;REPORT THE ERROR
2348 004702 000137 006066      JMP     5$                ;BYPASS REST OF THE SUBTEST
2349 004706
2350 004706 113760 001224 000010      64$:  MOVVB   PORTA,RPCS2(RO)   ;SELECT PORT A
2351 004714 013737 001224 001234      MOV     PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2352 004722 016037 000012 001126      MOV     RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2353 004730 012737 011600 001124      MOV     #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2354 004736 013737 001124 001166      MOV     $GDDAT,$TMP1     ;USE GOOD DATA AS A MASK
2355 004744 005137 001166      COM     $TMP1            ;COMPLEMENT THE EXPECTED STATUS
2356 004750 013737 001126 001164      MOV     $BDDAT,$TMP0     ;SAVE THE ACTUAL STATUS
2357 004756 043737 001166 001164      BIC     $TMP1,$TMP0      ;CLEAR UNWANTED BITS
2358 004764 023737 001124 001164      CMP     $GDDAT,$TMP0     ;ARE THE EXPECTED STATUS BITS SET ?
2359 004772 001401      BEQ     65$              ;BR IF THEY ARE
2360 004774 104005      ERROR  5                 ;REPORT THE ERROR
2361 004776 000240      65$:  NOP
2362
2363      ;:*****
2364      ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2365
2366 005000 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)   ;SELECT PORT B
2367 005006 013737 001226 001234      MOV     PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2368 005014 016046 000046      MOV     RPEC2(RO),-(SP)   ;STORE REGISTER RPEC2, PORT B, FOR CHECK
2369 005020 016046 000044      MOV     RPEC1(RO),-(SP)   ;STORE REGISTER RPEC1, PORT B, FOR CHECK
2370 005024 016046 000042      MOV     RPER3(RO),-(SP)   ;STORE REGISTER RPER3, PORT B, FOR CHECK
2371 005030 016046 000030      MOV     RPSN(RO),-(SP)    ;STORE REGISTER RPSN, PORT B, FOR CHECK
2372 005034 016046 000036      MOV     RPCC(RO),-(SP)    ;STORE REGISTER RPCC, PORT B, FOR CHECK
  
```

G04

```

2373 005040 016046 000034      MOV      RPCA(RO),-(SP)  ;STORE REGISTER RPCA, PORT B, FOR CHECK
2374 005044 016046 000032      MOV      RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT B, FOR CHECK
2375 005050 016046 000040      MOV      RPER2(RO),-(SP);STORE REGISTER RPER2, PORT B, FOR CHECK
2376 005054 016046 000020      MOV      RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT B, FOR CHECK
2377 005060 016046 000026      MOV      RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT B, FOR CHECK
2378 005064 016046 000006      MOV      RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT B, FOR CHECK
2379 005070 016046 000024      MOV      RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT B, FOR CHECK
2380 005074 016046 000014      MOV      RPER1(RO),-(SP);STORE REGISTER RPER1, PORT B, FOR CHECK
2381
2382                                     ;*****
2383                                     ;WAIT FOR PORT A TO TIMEOUT
2384
2385 005100 005760 000012      1$:      TST      RPDS1(RO)  ;WAIT FOR THE DRIVE TO TIMEOUT
2386 005104 001006 000000      BNE      2$           ;BR WHEN TIMEOUT OCCURS
2387 005106 005737 001254      TST      WATCH        ;CHECK WATCH
2388 005112 001372 000000      BNE      1$           ;BR IF NOT ZERO
2389 005114 104036 000000      ERROR    3$           ;NO TIMEOUT WITHIN 2 SECONDS
2390 005116 000137 005506      JMP      4$           ;BYPASS TIMEOUT TIME CHECK
2391 005122 012737 000340 177776 2$:      MOV      #(<7*32.>),@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2392 005130 013737 001252 001256      MOV      TIME,TIMEA    ;SAVE THE ELAPSED TIME FOR PORT A
2393 005136 004537 056630      JSR      RS,TOLER      ;CALCULATE THE TOLERANCE
2394 005142 001256 000000      .WORD   TIMEA         ;TIMEOUT VALUE FOR PORT A
2395 005144 012637 001260      MOV      (SP)+,TIMEAP  ;+25% TOLERANCE
2396 005150 012637 001262      MOV      (SP)+,TIMEAM  ;-25% TOLERANCE
2397
2398                                     ;*****
2399                                     ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2400
2401 005154 023727 001252 000764      CMP      TIME,#500.    ;WAS MEASURED TIME AT LEAST 500 MS?
2402 005162 103001 000000      BHIS     3$           ;BR IF IT WAS
2403 005164 104055 000000      ERROR    5$           ;REPORT TIMEOUT TOO SHORT
2404
2405                                     ;*****
2406                                     ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2407
2408 005166 012737 000240 177776 3$:      MOV      #(<5*32.>),@#PS ;RESTORE PRIORITY TO 5
2409
2410                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2411
2412 005174 005037 001250 000000      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
2413 005200 012737 000012 001122      MOV      #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2414 005206 060037 001122 000000      ADD      RO,$BDAOR    ;ADD THE I/O BASE ADDRESS
2415 005212 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$DDAT ;COMPARISON CONSTANT
2416 005220 113760 001224 000010      MOV      PORTA,RPCS2(RO);SELECT PORT A.
2417 005226 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2418 005234 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
2419 005242 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2420 005250 113760 001226 000010      MOV      PORTB,RPCS2(RO);SELECT PORT B.
2421 005256 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2422 005264 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
2423 005272 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2424 005300 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2425 005306 001006 000000      BNE      66$          ;BR IF NOT
2426 005310 005737 001164 000000      TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2427 005314 001037 000000      BNE      68$          ;BR IF NOT
2428 005316 104046 000000      ERROR    4$           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED

```

H04

```

2429 005320 000137 005504      JMP      70$      ;BYPASS THE REST OF THE CHECKS
2430 005324 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2431 005332 013737 001226 001234      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2432 005340 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
2433 005346 005737 001164      TST    $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
2434 005352 001414      BEQ    67$      ;BR IF ZERO
2435 005354 013737 001224 001234      MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2436 005362 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2437 005370 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
2438 005376 005737 001166      TST    $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
2439 005402 001004      BNE    68$      ;BR IF NOT
2440 005404 012737 177777 001250 67$:  MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR.
2441 005412 104022      ERROR  22        ;TYPE ERROR MESSAGE 22
2442 005414 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
2443 005422 013737 001224 001234      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
2444 005430 042737 100100 001170      BIC    #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
2445 005436 023737 001124 001170      CMP    $GDDAT,$TMP2 ;ALL BITS OK ?
2446 005444 001401      BEQ    69$      ;BR IF OK FROM PORT A.
2447 005446 104007      ERROR  7        ;REPORT ERROR
2448 005450 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2449 005456 013737 001226 001234      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
2450 005464 042737 100100 001172      BIC    #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
2451 005472 023737 001124 001172      CMP    $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
2452 005500 001401      BEQ    70$      ;BR IF OK
2453 005502 104007      ERROR  7        ;REPORT ERROR
2454 005504 000240 70$:  NOP

```

```

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

```

```

2455
2456
2457
2458
2459
2460 005506 013737 001226 001234 4$:  MOV     PORTB,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2461 005514 010037 001122      MOV     RO,$BDADR  ;BASE ADDRESS FOR REGISTER RPER1
2462 005520 062737 000014 001122      ADD    #RPER1,$BDADR ;ADDRESS OF RPER1 FOR TYPEOUT
2463 005526 012637 001126      MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
2464 005532 001401      BEQ    .+4        ;CONTENTS ZERO ?
2465 005534 104006      ERROR  6        ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2466 005536 010037 001122      MOV     RO,$BDADR  ;BASE ADDRESS FOR REGISTER RPMR
2467 005542 062737 000024 001122      ADD    #RPMR,$BDADR ;ADDRESS OF RPMR FOR TYPEOUT
2468 005550 012637 001126      MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPMR
2469 005554 001401      BEQ    .+4        ;CONTENTS ZERO ?
2470 005556 104006      ERROR  6        ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2471 005560 010037 001122      MOV     RO,$BDADR  ;BASE ADDRESS FOR REGISTER RPDA
2472 005564 062737 000006 001122      ADD    #RPDA,$BDADR ;ADDRESS OF RPDA FOR TYPEOUT
2473 005572 012637 001126      MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDA
2474 005576 001401      BEQ    .+4        ;CONTENTS ZERO ?
2475 005600 104006      ERROR  6        ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2476 005602 010037 001122      MOV     RO,$BDADR  ;BASE ADDRESS FOR REGISTER RPDT
2477 005606 062737 000026 001122      ADD    #RPDT,$BDADR ;ADDRESS OF RPDT FOR TYPEOUT
2478 005614 012637 001126      MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDT
2479 005620 001401      BEQ    .+4        ;CONTENTS ZERO ?
2480 005622 104006      ERROR  6        ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2481 005624 010037 001122      MOV     RO,$BDADR  ;BASE ADDRESS FOR REGISTER RPLA
2482 005630 062737 000020 001122      ADD    #RPLA,$BDADR ;ADDRESS OF RPLA FOR TYPEOUT
2483 005636 012637 001126      MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPLA
2484 005642 001401      BEQ    .+4        ;CONTENTS ZERO ?

```


2485	005644	104006			ERROR	6	;REPORT THAT PORT B SAW NON-ZERO REGISTER
2486	005646	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPER2
2487	005652	062737	000040	001122	ADD	#RPER2,\$BDADR	;ADDRESS OF RPER2 FOR TYPEOUT
2488	005660	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPER2
2489	005664	001401			BEQ	.+4	;CONTENTS ZERO ?
2490	005666	104006			ERROR	6	;REPORT THAT PORT B SAW NON-ZERO REGISTER
2491	005670	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPOF
2492	005674	062737	000032	001122	ADD	#RPOF,\$BDADR	;ADDRESS OF RPOF FOR TYPEOUT
2493	005702	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPOF
2494	005706	001401			BEQ	.+4	;CONTENTS ZERO ?
2495	005710	104006			ERROR	6	;REPORT THAT PORT B SAW NON-ZERO REGISTER
2496	005712	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPCA
2497	005716	062737	000034	001122	ADD	#RPCA,\$BDADR	;ADDRESS OF RPCA FOR TYPEOUT
2498	005724	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPCA
2499	005730	001401			BEQ	.+4	;CONTENTS ZERO ?
2500	005732	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2501	005734	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPCC
2502	005740	062737	000036	001122	ADD	#RPCC,\$BDADR	;ADDRESS OF RPCC FOR TYPEOUT
2503	005746	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPCC
2504	005752	001401			BEQ	.+4	;CONTENTS ZERO ?
2505	005754	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2506	005756	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPSN
2507	005762	062737	000030	001122	ADD	#RPSN,\$BDADR	;ADDRESS OF RPSN FOR TYPEOUT
2508	005770	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPSN
2509	005774	001401			BEQ	.+4	;CONTENTS ZERO ?
2510	005776	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2511	006000	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPER3
2512	006004	062737	000042	001122	ADD	#RPER3,\$BDADR	;ADDRESS OF RPER3 FOR TYPEOUT
2513	006012	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPER3
2514	006016	001401			BEQ	.+4	;CONTENTS ZERO ?
2515	006020	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2516	006022	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPEC1
2517	006026	062737	000044	001122	ADD	#RPEC1,\$BDADR	;ADDRESS OF RPEC1 FOR TYPEOUT
2518	006034	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPEC1
2519	006040	001401			BEQ	.+4	;CONTENTS ZERO ?
2520	006042	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2521	006044	010037	001122		MOV	RO,\$BDADR	;BASE ADDRESS FOR REGISTER RPEC2
2522	006050	062737	000046	001122	ADD	#RPEC2,\$BDADR	;ADDRESS OF RPEC2 FOR TYPEOUT
2523	006056	012637	001126		MOV	(SP)+,\$BDDAT	;CHECK THE STORED CONTENTS OF RPEC2
2524	006062	001401			BEQ	.+4	;CONTENTS ZERO ?
2525	006064	104006			ERROR	6	;REPORT THAT PORT B SEES NON-ZERO REGISTER
2526	006066	000004			SCOPE		;LOOP ?

55:

*TEST 3 PORT 'B' SEIZE/TIMEOUT TEST

*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES THE DRIVE.

* A. WRITE 0'S INTO RPS1 THROUGH PORT 'B'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.

* B. READ EACH DRIVE REGISTER, EXCEPT RPS1, THROUGH PORT 'A'; VERIFY THAT 0'S ARE READ FROM EACH REGISTER.

* C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.

2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540

```

2541      ;* MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
2542      ;* VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
2543      ;* NEUTRAL.
2544      ;*
2545      ;*****
2546      TST3:
2547      006070 005737 001274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
2548      006074 001406      BEQ      2$          ;BR IF NOT
2549      006076 100002      BPL      1$          ;BR IF JUST ENTERED TEST
2550      006100 000137 002602      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
2551      006104 012737 177777 001274 1$:      MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
2552      006112 112737 000003 001102 2$:      MOVVB   #3,$TSTNM   ;TEST NUMBER
2553      006120 012737 006142 001106      MOV      #TEST3,$LPADR ;LOAD LOOP ON TEST ADDRESS
2554      006126 012737 006142 001110      MOV      #TEST3,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2555      006134 012737 000012 001176      MOV      #10,$TIMES   ;DO 10. ITERATIONS
2556      006142 012706 001100      TEST3: MOV      #STACK,SP ;LOAD THE STACK POINTER
2557      006146 012737 000240 177776      MOV      #(<5*32.>),2#PS ;SET PRIORITY TO 5 IN CASE LOOPING
2558      006154 005037 001264      CLR      TIMEB       ;CLEAR TIMEOUT VALUE FOR PORT B
2559      006160 005037 001266      CLR      TIMEBP      ;CLEAR UPPER TIMEOUT TOLERANCE
2560      006164 005037 001270      CLR      TIMEBM      ;CLEAR LOWER TIMEOUT TOLERANCE
2561
2562      ;*****
2563      ;START THE TIMER
2564
2565      006170 005037 001252      CLR      TIME         ;CLEAR THE ELAPSED TIME COUNTER
2566      006174 012737 003720 001254      MOV      #2000.,WATCH ;SET WATCH TO 2000 MS
2567
2568      ;*****
2569
2570      ;SEIZE THE DRIVE THROUGH PORT B
2571
2572      006202 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
2573      006210 013737 001226 001236      MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
2574      006216 005060 000012      CLR      RPDS1(RO)     ;WRITE RPDS1
2575      006222 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
2576      006230 013737 001224 001234      MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2577      006236 013737 001224 001240      MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
2578      006244 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
2579      006252 010037 001122      MOV      RO,$BDADR     ;R#11 BASE ADDRESS
2580      006256 062737 000012 001122      ADD      #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
2581      006264 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
2582      006270 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
2583      006276 001403      BEQ      64$          ;BR IF IT IS
2584      006300 104004      ERROR   4             ;REPORT THE ERROR
2585      006302 000137 007466      JMP      5$           ;BYPASS REST OF THE SUBTEST
2586      006306
2587      006306 113760 001226 000010      64$:  MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
2588      006314 013737 001226 001234      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2589      006322 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2590      006330 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2591      006336 013737 001124 001166      MOV      $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
2592      006344 005137 001166      COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
2593      006350 013737 001126 001164      MOV      $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
2594      006356 043737 001166 001164      BIC      $TMP1,$TMP0   ;CLEAR UNWANTED BITS
2595      006364 023737 001124 001164      CMP      $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
2596      006372 001401      BEQ      65$          ;BR IF THEY ARE

```

K04

```

2597 006374 104005          ERROR 5          ;REPORT THE ERROR
2598 006376 000240        65$: NOP
2599
2600
2601          ;*****
2602          ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2603 006400 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
2604 006406 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2605 006414 016046 000046          MOV    RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
2606 006420 016046 000044          MOV    RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
2607 006424 016046 000042          MOV    RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
2608 006430 016046 000030          MOV    RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT A, FOR CHECK
2609 006434 016046 000036          MOV    RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT A, FOR CHECK
2610 006440 016046 000034          MOV    RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT A, FOR CHECK
2611 006444 016046 000032          MOV    RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT A, FOR CHECK
2612 006450 016046 000040          MOV    RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
2613 006454 016046 000020          MOV    RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT A, FOR CHECK
2614 006460 016046 000026          MOV    RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT A, FOR CHECK
2615 006464 016046 000006          MOV    RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT A, FOR CHECK
2616 006470 016046 000024          MOV    RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT A, FOR CHECK
2617 006474 016046 000014          MOV    RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK
2618
2619          ;*****
2620          ;WAIT FOR PORT B TO TIMEOUT
2621
2622 006500 005760 000012        1$: TST    RPDS1(RO)    ;WAIT FOR THE DRIVE TO TIMEOUT
2623 006504 001006          BNE    2$           ;BR WHEN TIMEOUT OCCURS
2624 006506 005737 001254          TST    WATCH        ;CHECK WATCH
2625 006512 001372          BNE    1$           ;BR IF NOT ZERO
2626 006514 104036          ERROR  36          ;NO TIMEOUT WITHIN 2 SECONDS
2627 006516 000137 007106          JMP    4$           ;BYPASS TIMEOUT TIME CHECK
2628 006522 012737 000340 177776 2$: MOV    #'*32.),@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2629 006530 013737 001252 001264      MOV    TIME,TIMEB   ;SAVE THE ELAPSED TIME FOR PORT B
2630 006536 004537 056630          JSR    R5,TOLER     ;CALCULATE THE TOLERANCE
2631 006542 001264          .WORD  TIMEB       ;TIMEOUT VALUE FOR PORT B
2632 006544 012637 001266          MOV    (SP)+,TIMEBP ;+25% TOLERANCE
2633 006550 012637 001270          MOV    (SP)+,TIMEBM ; -25% TOLERANCE
2634
2635          ;*****
2636          ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2637
2638 006554 023727 001252 000764          CMP    TIME,#500.   ;WAS MEASURED TIME AT LEAST 500 MS?
2639 006562 103001          BHS    3$           ;BR IF IT WAS
2640 006564 104055          ERROR  55          ;REPORT TIMEOUT TOO SHORT
2641
2642          ;*****
2643          ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT
2644
2645 006566 012737 000240 177776 3$: MOV    #'*32.),@#PS ;RESTORE PRIORITY TO 5
2646
2647          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2648
2649 006574 005037 001250          CLR    RELERR       ;CLEAR THE 'RELEASE ERROR ' INDICATOR
2650 006600 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2651 006606 060037 001122          ADD    RO,$BDADR    ;ADD THE I/O BASE ADDRESS
2652 006612 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
  
```

```

2653 006620 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
2654 006626 016037 000012 001170      MOV      RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2655 006634 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
2656 006642 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2657 006650 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
2658 006656 016037 000012 001172      MOV      RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2659 006664 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
2660 006672 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2661 006700 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2662 006706 001006                BNE      66$ ;BR IF NOT
2663 006710 005737 001164                TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2664 006714 001037                BNE      68$ ;BR IF NOT
2665 006716 104046                ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2666 006720 000137 007104                JMP      70$ ;BYPASS THE REST OF THE CHECKS
2667 006724 013737 001170 001126 66$:      MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2668 006732 013737 001226 001234                MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2669 006740 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
2670 006746 005737 001164                TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2671 006752 001414                BEQ      67$ ;BR IF ZERO
2672 006754 013737 001224 001234                MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2673 006762 013737 001172 001126                MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2674 006770 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
2675 006776 005737 001166                TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2676 007002 001004                BNE      68$ ;BR IF NOT
2677 007004 012737 177777 001250 67$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
2678 007012 104022                ERROR    22 ;TYPE ERROR MESSAGE 22
2679 007014 013737 001170 001126 68$:      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
2680 007022 013737 001224 001234                MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
2681 007030 042737 100100 001170      BIC      #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
2682 007036 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
2683 007044 001401                BEQ      69$ ;BR IF OK FROM PORT A.
2684 007046 104007                ERROR    7 ;REPORT ERROR
2685 007050 013737 001172 001126 69$:      MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2686 007056 013737 001226 001234                MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
2687 007064 042737 100100 001172      BIC      #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
2688 007072 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
2689 007100 001401                BEQ      70$ ;BR IF OK
2690 007102 104007                ERROR    7 ;REPORT ERROR
2691 007104 000240 70$:      NOP
2692
2693 ;:*****
2694 ;CHECK THE REGISTERS STORED THROUGH PORT A. ALL REGISTERS SHOULD BE ZERO.
2695 ;THE REGISTERS ARE STORED ON THE STACK.
2696
2697 007106 013737 001224 001234 4$:      MOV      PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2698 007114 010037 001122                MOV      RO,$BDADR ;BASE ADDRESS FOR REGISTER RPER1
2699 007120 062737 000014 001122                ADD      #RPER1,$BDADR ;ADDRESS OF RPER1 FOR TYPEOUT
2700 007126 012637 001126                MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
2701 007132 001401                BEQ      .+4 ;CONTENTS ZERO ?
2702 007134 104006                ERROR    6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2703 007136 010037 001122                MOV      RO,$BDADR ;BASE ADDRESS FOR REGISTER RPMR
2704 007142 062737 000024 001122                ADD      #RPMR,$BDADR ;ADDRESS OF RPMR FOR TYPEOUT
2705 007150 012637 001126                MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPMR
2706 007154 001401                BEQ      .+4 ;CONTENTS ZERO ?
2707 007156 104006                ERROR    6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2708 007160 010037 001122                MOV      RO,$BDADR ;BASE ADDRESS FOR REGISTER RPOA

```

2709	007164	062737	000006	001122	ADD	#RPDA,\$BDADR	; ADDRESS OF RPDA FOR TYPEOUT
2710	007172	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPDA
2711	007176	001401			BEQ	.+4	; CONTENTS ZERO ?
2712	007200	104006			ERROR	6	; REPORT THAT PORT A SAW NON-ZERO REGISTER
2713	007202	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPDT
2714	007206	062737	000026	001122	ADD	#RPDT,\$BDADR	; ADDRESS OF RPDT FOR TYPEOUT
2715	007214	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPDT
2716	007220	001401			BEQ	.+4	; CONTENTS ZERO ?
2717	007222	104006			ERROR	6	; REPORT THAT PORT A SAW NON-ZERO REGISTER
2718	007224	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPLA
2719	007230	062737	000020	001122	ADD	#RPLA,\$BDADR	; ADDRESS OF RPLA FOR TYPEOUT
2720	007236	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPLA
2721	007242	001401			BEQ	.+4	; CONTENTS ZERO ?
2722	007244	104006			ERROR	6	; REPORT THAT PORT A SAW NON-ZERO REGISTER
2723	007246	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPER2
2724	007252	062737	000040	001122	ADD	#RPER2,\$BDADR	; ADDRESS OF RPER2 FOR TYPEOUT
2725	007260	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPER2
2726	007264	001401			BEQ	.+4	; CONTENTS ZERO ?
2727	007266	104006			ERROR	6	; REPORT THAT PORT A SAW NON-ZERO REGISTER
2728	007270	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPOF
2729	007274	062737	000032	001122	ADD	#RPOF,\$BDADR	; ADDRESS OF RPOF FOR TYPEOUT
2730	007302	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPOF
2731	007306	001401			BEQ	.+4	; CONTENTS ZERO ?
2732	007310	104006			ERROR	6	; REPORT THAT PORT A SAW NON-ZERO REGISTER
2733	007312	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPCA
2734	007316	062737	000034	001122	ADD	#RPCA,\$BDADR	; ADDRESS OF RPCA FOR TYPEOUT
2735	007324	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPCA
2736	007330	001401			BEQ	.+4	; CONTENTS ZERO ?
2737	007332	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2738	007334	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPCC
2739	007340	062737	000036	001122	ADD	#RPCC,\$BDADR	; ADDRESS OF RPCC FOR TYPEOUT
2740	007346	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPCC
2741	007352	001401			BEQ	.+4	; CONTENTS ZERO ?
2742	007354	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2743	007356	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPSN
2744	007362	062737	000030	001122	ADD	#RPSN,\$BDADR	; ADDRESS OF RPSN FOR TYPEOUT
2745	007370	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPSN
2746	007374	001401			BEQ	.+4	; CONTENTS ZERO ?
2747	007376	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2748	007400	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPER3
2749	007404	062737	000042	001122	ADD	#RPER3,\$BDADR	; ADDRESS OF RPER3 FOR TYPEOUT
2750	007412	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPER3
2751	007416	001401			BEQ	.+4	; CONTENTS ZERO ?
2752	007420	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2753	007422	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPEC1
2754	007426	062737	000044	001122	ADD	#RPEC1,\$BDADR	; ADDRESS OF RPEC1 FOR TYPEOUT
2755	007434	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPEC1
2756	007440	001401			BEQ	.+4	; CONTENTS ZERO ?
2757	007442	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2758	007444	010037	001122		MOV	RO,\$BDADR	; BASE ADDRESS FOR REGISTER RPEC2
2759	007450	062737	000046	001122	ADD	#RPEC2,\$BDADR	; ADDRESS OF RPEC2 FOR TYPEOUT
2760	007456	012637	001126		MOV	(SP)+,\$BDDAT	; CHECK THE STORED CONTENTS OF RPEC2
2761	007462	001401			BEQ	.+4	; CONTENTS ZERO ?
2762	007464	104006			ERROR	6	; REPORT THAT PORT A SEES NON-ZERO REGISTER
2763	007466	000004			SS:	SCOPE	; LOOP ?
2764							

2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820

007470
007470 005737 001274
007470 001406
007474 100002
007476 000002
007500 000137 002602
007504 012737 177777 001274
007512 112737 000004 001102
007520 012737 007542 001106
007526 012737 007542 001110
007534 012737 000001 001176
007542 012706 001100
007546 113760 001224 000010
007554 013737 001224 001234

007562 005037 001252
007566 012737 003720 001254
007574 013737 001224 001236

007610 113760 001226 000010
007616 013737 001226 001234
007624 005037 001244
007630 016037 000012 001126
007636 012737 000012 001122
007644 060037 001122
007650 005037 001124
007654 023737 001124 001126

```
*****  
:TEST 4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'  
:  
:VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'  
:FOR THE PORT UNDER TEST.  
:  
: A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE  
: DRIVE WAS SEIZED BY PORT 'A' AND THAT THE 'GO' BIT RESET.  
:  
: B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE  
: 'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET  
: FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT  
: 'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)  
:  
: C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED  
: THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.  
:  
*****  
:START THE TIMER  
:  
:PERFORMING ONLY SINGLE TESTS ?  
:BR IF NOT  
:BR IF JUST ENTERED TEST  
:RETURN & GET NEXT TEST NUMBER  
:SET SINGLE TEST INDICATOR  
:TEST NUMBER  
:LOAD LOOP ON TEST ADDRESS  
:LOAD LOOP ON ERROR ADDRESS  
:DO 1 ITERATION  
:LOAD THE STACK POINTER  
:SELECT PORT A  
:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
:  
:ISSUE DRIVE CLEAR COMMAND  
:  
:ISSUE A DRIVE CLEAR  
:  
:VERIFY THAT DRIVE SEIZED BY PORT A.  
:  
:SELECT PORT B  
:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
:CLEAR THE 'CHECK ERROR' INDICATOR  
:GET CONTENTS OF RPDS1  
:FORM REGISTER ADDRESS OF ERROR MESSAGE  
:ADD RH11 BASE ADDRESS  
:WHAT REGISTER SHOULD BE  
:IS THE REGISTER OK ?
```

B05

```

2821 007662 001403          BEQ      64$          ;BR IF OK
2822 007664 104012          ERROR    12          ;TYPE MESSAGE 12
2823 007666 005137 001244        COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
2824 007672 000240          NOP
2825 007674 113760 001224 000010 64$:  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
2826 007702 013737 001224 001234  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2827 007710 005037 001244          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
2828 007714 016037 000012 001126  MOV     RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2829 007722 012737 000012 001122  MOV     #RPDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2830 007730 060037 001122          ADD     RO,SBADR     ;ADD RH11 BASE ADDRESS
2831 007734 012737 011600 001124  MOV     #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
2832 007742 013737 001126 001164  MOV     SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
2833 007750 042737 106177 001164  BIC     #1C71600,$TMP0 ;SAVE SPECIFIED BITS
2834 007756 023737 001124 001164  CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
2835 007764 001414          BEQ     66$
2836 007766 013737 001126 001174  MOV     SBDDAT,$TMP4  ;COPY 'BAD DATA'
2837 007774 042737 071600 001174  BIC     #71600,$TMP4  ;CLEAR THE MASKED BITS
2838 010002 053737 001174 001124  BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
2839 010010 104010          ERROR    10          ;REPORT THE ERROR
2840 010012 005137 001244        COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
2841 010016 000240          NOP
2842 010020 005037 001244          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
2843 010024 016037 000000 001126  MOV     RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
2844 010032 012737 000000 001122  MOV     #RPCS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2845 010040 060037 001122          ADD     RO,SBADR     ;ADD RH11 BASE ADDRESS
2846 010044 012737 004210 001124  MOV     #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
2847 010052 013737 001126 001164  MOV     SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
2848 010060 042737 100000 001164  BIC     #1C77777,$TMP0 ;SAVE SPECIFIED BITS
2849 010066 023737 001124 001164  CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
2850 010074 001414          BEQ     68$
2851 010076 013737 001126 001174  MOV     SBDDAT,$TMP4  ;COPY 'BAD DATA'
2852 010104 042737 077777 001174  BIC     #77777,$TMP4  ;CLEAR THE MASKED BITS
2853 010112 053737 001174 001124  BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
2854 010120 104010          ERROR    10          ;REPORT THE ERROR
2855 010122 005137 001244        COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
2856 010126 000240          NOP
2857
2858 ;:*****
2859 ;ISSUE READIN PRESET COMMAND AND SET FMT22
2860
2861 010130 012760 000023 000000  MOV     #23,RPCS1(RO) ;ISSUE A READIN PRESET
2862 010136 012760 010000 000032  MOV     #FMT22,RPOF(RO) ;SET FMT22
2863
2864 ;:*****
2865 ;VERIFY THAT THE DRIVE STATUS IS CORRECT
2866
2867 010144 005037 001244          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
2868 010150 016037 000012 001126  MOV     RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2869 010156 012737 000012 001122  MOV     #RPDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2870 010164 060037 001122          ADD     RO,SBADR     ;ADD RH11 BASE ADDRESS
2871 010170 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
2872 010176 013737 001126 001164  MOV     SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
2873 010204 042737 106077 001164  BIC     #1C71700,$TMP0 ;SAVE SPECIFIED BITS
2874 010212 023737 001124 001164  CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
2875 010220 001414          BEQ     70$
2876 010222 013737 001126 001174  MOV     SBDDAT,$TMP4  ;COPY 'BAD DATA'

```

C05

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 54
 DZRJE.A.CMB 02-NOV-76 18:40 T4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'

```

2977 010230 042737 071700 001174      BIC      #71700,$STMP4      ;CLEAR THE MASKED BITS
2978 010236 053737 001174 001124      BIS      $STMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2979 010244 104013                ERROR    13                ;TYPE MESSAGE 13
2880 010246 005137 001244                COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2881 010252 000240                NOP
2882 010254 113760 001226 000010      MOVB     PORTB,RPCS2(RO)  ;SELECT PORT B
2883 010262 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2884
2885 ;:*****
2886 ;WAIT FOR TIMEOUT TO RELEASE DRIVE
2887
2888 010270 005760 000012      15:     TST      RPDS1(RO)    ;WAIT FOR THE PORT TO TIME OUT
2889 010274 001006                BNE     25                ;BR WHEN TIMEOUT OCCURS
2890 010276 005737 001254                TST     WATCH            ;CHECK THE WATCH
2891 010302 001372                BNE     15                ;BR IF NOT ZERO
2892 010304 104036                ERROR   36                ;NO TIMEOUT WITHIN 2 SECONDS
2893 010306 000137 010624                JMP     35                ;BYPASS ATTN REGISTER CHECK
2894
2895 ;:*****
2896 ;SEE IF DRIVE RETURNED TO NEUTRAL
2897
2898 010312      25:
2899
2900 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2901
2902 010312 005037 001250      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
2903 010316 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2904 010324 060037 001122                ADD     RO,$BDADR        ;ADD THE I/O BASE ADDRESS
2905 010330 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2906 010336 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
2907 010344 016037 000012 001170      MOV      RPDS1(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2908 010352 013737 001170 001164      MOV      $STMP2,$STMP0   ;COPY IT INTO 'STMP0'
2909 010360 042737 100100 001164      BIC      #ATA!VV,$STMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2910 010366 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
2911 010374 016037 000012 001172      MOV      RPDS1(RO),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2912 010402 013737 001172 001166      MOV      $STMP3,$STMP1  ;COPY IT INTO 'STMP1'
2913 010410 042737 100100 001166      BIC      #ATA!VV,$STMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2914 010416 023737 001164 001166      CMP      $STMP0,$STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2915 010424 001006                BNE     72$              ;BR IF NOT
2916 010426 005737 001164                TST     $STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2917 010432 001037                BNE     74$              ;BR IF NOT
2918 010434 104046                ERROR   46                ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2919 010436 000137 010622                JMP     76$              ;BYPASS THE REST OF THE CHECKS
2920 010442 013737 001170 001126 72$:     MOV      $STMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2921 010450 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2922 010456 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
2923 010464 005737 001164                TST     $STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2924 010470 001414                BEQ     73$              ;BR IF ZERO
2925 010472 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2926 010500 013737 001172 001126      MOV      $STMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
2927 010506 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
2928 010514 005737 001166                TST     $STMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2929 010520 001004                BNE     74$              ;BR IF NOT
2930 010522 012737 177777 001250 73$:     MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2931 010530 104022                ERROR   22                ;TYPE ERROR MESSAGE 22
2932 010532 013737 001170 001126 74$:     MOV      $STMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ

```



```

2933 010540 013737 001224 001234      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
2934 010546 042737 100100 001170      BIC      #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2935 010554 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
2936 010562 001401                BEQ      75$              ;BR IF OK FROM PORT A.
2937 010564 104007                ERROR    7                ;REPORT ERROR
2938 010566 013737 001172 001126 75$:    MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2939 010574 013737 001226 001234      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
2940 010602 042737 100100 001172      BIC      #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2941 010610 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2942 010616 001401                BEQ      76$              ;BR IF OK
2943 010620 104007                ERROR    7                ;REPORT ERROR
2944 010622 000240                NOP
2945 010624 000004                3$:      SCOPE              ;LOOP ?

```

*TEST 5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
* FOR THE PORT UNDER TEST.

- * A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE WAS SEIZED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
- * B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'VV' BIT FOR PORT 'B' WAS SET.
- * C. STALL FOR 2 SECCNDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.

*TESTS:

```

2963 010626                TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2964 010626 005737 001274                BEQ      2$              ;BR IF NOT
2965 010632 001406                BPL      1$              ;BR IF JUST ENTERED TEST
2966 010634 100002                JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
2967 010636 000137 002602                MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
2968 010642 012737 177777 001274 1$:    MOV      #5,$STSTM      ;TEST NUMBER
2969 010650 112737 000005 001102 2$:    MOV      #TEST5,$LPAOR  ;LOAD LOOP ON TEST ADDRESS
2970 010656 012737 010700 001106      MOV      #TEST5,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2971 010664 012737 010700 001110      MOV      #1,$TIMES     ;DO 1 ITERATION
2972 010672 012737 000001 001176      MOV      #STACK,$SP    ;LOAD THE STACK POINTER
2973 010700 012706 001100                MOV      PORTB,$PC52(R0);SELECT PORT B
2974 010704 113760 001226 000010      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2975 010712 013737 001226 001234

```

*START THE TIMER

```

2979                CLR      TIME            ;CLEAR THE ELAPSED TIME COUNTER
2980 010720 005037 001252                MOV      #2000,$WATCH  ;SET WATCH TO 2000 MS
2981 010724 012737 003720 001254      MOV      PORTB,$SEIZPT ;'SEIZED' PORT ADDRESS
2982 010732 013737 001226 001236

```

*ISSUE DRIVE CLEAR COMMAND

```

2987 010740 012760 000011 000000      MOV      #11,$RPCS1(R0);ISSUE A DRIVE CLEAR
2988

```

E05

```

2989
2990
2991
2992 010746 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
2993 010754 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2994 010762 005037 001244          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2995 010766 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2996 010774 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2997 011002 060037 001122          ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
2998 011006 005037 001124          CLR    $GDDAT ;WHAT REGISTER SHOULD BE
2999 011012 023737 001124 001126      CMP    $GDDAT,SBDDAT ;IS THE REGISTER OK ?
3000 011020 001403          BEQ   64$ ;BR IF OK
3001 011022 104012          ERROR 12 ;TYPE MESSAGE 12
3002 011024 005137 001244          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3003 011030 000240          NOP
3004 011032 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
3005 011040 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3006 011046 005037 001244          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3007 011052 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
3008 011060 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3009 011066 060037 001122          ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
3010 011072 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
3011 011100 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3012 011106 042737 106177 001164      BIC   #1C71600,$TMP0 ;SAVE SPECIFIED BITS
3013 011114 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
3014 011122 001414          BEQ   66$ ;BR IF OK
3015 011124 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
3016 011132 042737 071600 001174      BIC   #71600,$TMP4 ;CLEAR THE MASKED BITS
3017 011140 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3018 011146 104010          ERROR 10 ;REPORT THE ERROR
3019 011150 005137 001244          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3020 011154 000240          NOP
3021 011156 005037 001244          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3022 011162 016037 000000 001126      MOV    RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
3023 011170 012737 000000 001122      MOV    #RPCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3024 011176 060037 001122          ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
3025 011202 012737 004210 001124      MOV    #4210,$GDDAT ;WHAT REGISTER SHOULD BE
3026 011210 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3027 011216 042737 100000 001164      BIC   #1C77777,$TMP0 ;SAVE SPECIFIED BITS
3028 011224 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
3029 011232 001414          BEQ   68$ ;BR IF OK
3030 011234 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
3031 011242 042737 077777 001174      BIC   #77777,$TMP4 ;CLEAR THE MASKED BITS
3032 011250 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3033 011256 104010          ERROR 10 ;REPORT THE ERROR
3034 011260 005137 001244          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3035 011264 000240          NOP
3036
3037
3038
3039
3040 011266 012760 000023 000000      MOV    #23,RPCS1(RO) ;ISSUE A READIN PRESET
3041 011274 012760 010000 000032      MOV    #FMT22,RPOF(RO) ;SET FMT22
3042
3043
3044

```

;VERIFY THAT DRIVE SEIZED BY PORT B.

64\$:

66\$:

68\$:

;ISSUE READIN PRESET COMMAND AND SET FMT22

;VERIFY THAT THE DRIVE STATUS IS CORRECT

F05

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 57
 DZRJEA.CMB 02-NOV-76 18:40 T5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

```

3045
3046 011302 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3047 011306 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3048 011314 012737 000012 001122 MOV #RPDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3049 011322 060037 001122 ADD RO, $BDADR ;ADD RHI1 BASE ADDRESS
3050 011326 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
3051 011334 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3052 011342 042737 106077 001164 BIC #1C71700, $TMP0 ;SAVE SPECIFIED BITS
3053 011350 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
3054 011356 001414 BEQ 70$ ;BR IF OK
3055 011360 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
3056 011366 042737 071700 001174 BIC #71700, $TMP4 ;CLEAR THE MASKED BITS
3057 011374 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3058 011402 104013 ERROR 13 ;TYPE MESSAGE 13
3059 011404 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3060 011410 000240 70$: NOP
3061 011412 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT A
3062 011420 013737 001224 001234 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3063
3064 ;*****
3065 ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3066
3067 011426 005760 000012 1$: TST RPDS1(RO) ;WAIT FOR THE PORT TO TIME OUT
3068 011432 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
3069 011434 005737 001254 TST WATCH ;CHECK THE WATCH
3070 011440 001372 BNE 1$ ;BR IF NOT ZERO
3071 011442 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
3072 011444 000137 011762 JMP 3$ ;BYPASS ATTN REGISTER CHECK
3073
3074 ;*****
3075 ;SEE IF DRIVE RETURNED TO NEUTRAL
3076
3077 011450 2$:
3078
3079 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3080
3081 011450 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3082 011454 012737 000012 001122 MOV #RPDS1, $BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3083 011462 060037 001122 ADD RO, $BDADR ;ADD THE I/O BASE ADDRESS
3084 011466 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3085 011474 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT A.
3086 011502 016037 000012 001170 MOV RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3087 011510 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
3088 011516 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3089 011524 113760 001226 000010 MOVB PORTB, RPCS2(RO) ;SELECT PORT B.
3090 011532 016037 000012 001172 MOV RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3091 011540 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
3092 011546 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3093 011554 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3094 011562 001006 BNE 72$ ;BR IF NOT
3095 011564 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3096 011570 001037 BNE 74$ ;BR IF NOT
3097 011572 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3098 011574 000137 011760 JMP 76$ ;BYPASS THE REST OF THE CHECKS
3099 011600 013737 001170 001126 72$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3100 011606 013737 001226 001234 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

G05

MD-11-DZJJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 58
 DZJJE.CMB 02-NOV-76 18:40 TS PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

```

3101 011614 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B.
3102 011622 005737 001164              TST  $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
3103 011626 001414              BEQ  73$              ;BR IF ZERO
3104 011630 013737 001224 001234      MOV  PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3105 011636 013737 001172 001126      MOV  $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
3106 011644 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A.
3107 011652 005737 001166              TST  $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
3108 011656 001004              BNE  74$              ;BR IF NOT
3109 011660 012737 177777 001250 73$:  MOV  #-1,RELEERR      ;SET 'RELEA-- ERROR' INDICATOR
3110 011656 104026              ERROR 26              ;TYPE ERROR MESSAGE 26
3111 011670 013737 001170 001126 74$:  MOV  $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3112 011676 013737 001224 001234      MOV  PORTA,PTNBR      ;CHANGE PORT NUMBER
3113 011704 042737 100000 001170      BIC  #ATA,$TMP2        ;DON'T CHECK THE ATTN BIT
3114 011712 023737 001124 001170      CMP  $GDDAT,$TMP2     ;ALL BITS OK ?
3115 011720 001401              BEQ  75$              ;BR IF OK FROM PORT A.
3116 011722 104007              ERROR 7                ;REPORT ERROR
3117 011724 013737 001172 001126 75$:  MOV  $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3118 011732 013737 001226 001234      MOV  PORTB,PTNBR      ;CHANGE PORT NUMBER
3119 011740 042737 100000 001172      BIC  #ATA,$TMP3        ;DON'T CHECK THE ATTN BIT
3120 011746 023737 001124 001172      CMP  $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
3121 011754 001401              BEQ  76$              ;BR IF OK
3122 011756 104007              ERROR 7                ;REPORT ERROR
3123 011760 000240 76$:  NOP
3124 011762 000004 3$:  SCOPE                ;LOOP ?

```

3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137

```

*****
*TEST 6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'
*
*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
* DRIVE.
*****

```

3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150

```

*****
TST6:
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 #6,$STNM       ;TEST NUMBER
MOV  #TEST6,$LPADR     ;LOAD LOOP ON TEST ADDRESS
MOV  #TEST6,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
MOV  #4000.,$TIMES     ;DO 4000. ITERATIONS
TEST6: MOV #STACK,$SP  ;LOAD THE STACK POINTER

```

3151
3152
3153
3154
3155
3156

```

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

```

H05

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 59
DZRJEA.CMB 02-NOV-76 18:40 T6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

```
3157 ;*****
3158
3159 ;SEIZE THE DRIVE THROUGH PORT A
3160
3161 012054 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
3162 012062 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
3163 012070 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
3164 012074 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
3165
3166 ;*****
3167
3168 ;RELEASE THE DRIVE FROM PORT A
3169
3170 012102 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
3171 012110 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3172 012116 012760 000013 000000 MOV #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A
3173
3174 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3175
3176 012124 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3177 012130 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3178 012136 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
3179 012142 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3180 012150 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
3181 012156 016037 000012 001170 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3182 012164 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
3183 012172 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3184 012200 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
3185 012206 016037 000012 001172 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3186 012214 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
3187 012222 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3188 012230 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3189 012236 001006 BNE 66$ ;BR IF NOT
3190 012240 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3191 012244 001037 BNE 68$ ;BR IF NOT
3192 012246 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3193 012250 000137 012434 JMP 70$ ;BYPASS THE REST OF THE CHECKS
3194 012254 013737 001170 001126 66$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3195 012262 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3196 012270 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
3197 012276 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3198 012302 001414 BEQ 67$ ;BR IF ZERO
3199 012304 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3200 012312 013737 001172 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3201 012320 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
3202 012326 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3203 012332 001004 BNE 68$ ;BR IF NOT
3204 012334 012737 177777 001250 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3205 012342 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
3206 012344 013737 001170 001126 68$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3207 012352 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
3208 012360 042737 100000 001170 BIC #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
3209 012366 023737 001124 001170 CMP $GDDAT,STMP2 ;ALL BITS OK ?
3210 012374 001401 BEQ 69$ ;BR IF OK FROM PORT A.
3211 012376 104007 ERROR 7 ;REPORT ERROR
3212 012400 013737 001172 001126 69$: MOV STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
```

```

3213 012406 013737 001226 001234      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
3214 012414 042737 100000 001172      BIC      #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3215 012422 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3216 012430 001401          BEQ      70$             ;BR IF OK
3217 012432 104007          ERROR    7              ;REPORT ERROR
3218 012434 000240          NOP
3219 012436 005737 001250      TST      RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
3220 012442 001402          BEQ      .+6            ;BR IF IN NEUTRAL
3221 012444 000137 012720      JMP      1$             ;GO WAIT FOR DRIVE TO TIMEOUT
3222 012450 113760 001224 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT A
3223 012456 013737 001224 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3224 012464 005037 001244      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
3225 012470 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
3226 012476 012737 000012 001122      MOV      #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3227 012504 060037 001122      ADD      RO,$BDADR      ;ADD R#11 BASE ADDRESS
3228 012510 005037 001124      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
3229 012514 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
3230 012522 042737 077777 001164      BIC      #1CATA,$TMP0   ;SAVE SPECIFIED BITS
3231 012530 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
3232 012536 001414          BEQ      71$            ;BR IF OK
3233 012540 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
3234 012546 042737 100000 001174      BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
3235 012554 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
3236 012562 104017          ERROR    17            ;TYPE MESSAGE 17
3237 012564 005137 001244      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
3238 012570 000240          NOP
3239 012572 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
3240 012600 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3241 012606 005037 001244      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
3242 012612 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
3243 012620 012737 000012 001122      MOV      #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3244 012626 060037 001122      ADD      RO,$BDADR      ;ADD #11 BASE ADDRESS
3245 012632 005037 001124      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
3246 012636 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
3247 012644 042737 077777 001164      BIC      #1CATA,$TMP0   ;SAVE SPECIFIED BITS
3248 012652 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
3249 012660 001414          BEQ      73$            ;BR IF OK
3250 012662 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
3251 012670 042737 100000 001174      BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
3252 012676 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
3253 012704 104017          ERROR    17            ;TYPE MESSAGE 17
3254 012706 005137 001244      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
3255 012712 000240          NOP
3256 012714 000137 012752      JMP      2$             ;GO CHECK FOR LOOP ON ERROR
3257
3258 ;:*****
3259 ;:IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3260 ;:TO RELEASE THE DRIVE
3261
3262 012720          1$:
3263 012720 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
3264 012726 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3265 012734 005760 000012      TST      RPDS1(RO)      ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3266 012740 001004          BNE     2$             ;BR WHEN DRIVE RELEASED
3267 012742 005737 001254      TST      WATCH          ;CHECK THE WATCH
3268 012746 001364          BNE     1$             ;BR IF NOT ZERO

```

J05

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 61
DZRJEA.CMB 02-NOV-76 18:40 T6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

```
3269 012750 104036          ERROR 36          ;NO TIMEOUT WITHIN 2 SECONDS
3270 012752 000004          2$: SCOPE          ;LOOP ?
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284 012754
3285 012754 005737 001274          TST  KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3286 012760 001406          BEQ  2$             ;BR IF NOT
3287 012762 100002          BPL  1$             ;BR IF JUST ENTERED TEST
3288 012764 000137 002602          JMP  EXEC           ;RETURN & GET NEXT TEST NUMBER
3289 012770 012737 177777 001274 1$: MOV  #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
3290 012776 112737 000007 001102 2$: MOVB #7,$STNM     ;TEST NUMBER
3291 013004 012737 013026 001106          MOV  #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
3292 013012 012737 013026 001110          MOV  #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3293 013020 012737 007640 001176          MOV  #4000,$TIMES  ;DO 4000. ITERATIONS
3294 013026 012706 001100          TEST7: MOV #STACK,SP ;LOAD THE STACK POINTER
3295
3296
3297
3298
3299
3300 013032 005037 001252          CLR  TIME           ;CLEAR THE ELAPSED TIME COUNTER
3301 013036 012737 003720 001254          MOV  #2000.,WATCH  ;SET WATCH TO 2000 MS
3302
3303
3304
3305
3306 013044 113760 001226 000010          MOVB PORTB,RPCS2(RO) ;SELECT PORT B
3307 013052 013737 001226 001236          MOV  PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
3308 013060 005060 000012          CLR  RPDS1(RO)     ;WRITE RPDS1
3309 013064 013737 001224 001240          MOV  PORTA,OPRPT   ;'OPPOSITE' PORT ADDRESS
3310
3311
3312
3313
3314
3315 013072 113760 001226 000010          MOVB PORTB,RPCS2(RO) ;SELECT PORT B
3316 013100 013737 001226 001234          MOV  PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3317 013106 012760 000013 000000          MOV  #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3318
3319
3320
3321
3322
3323
3324 013114 005037 001250          CLR  RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3322 013120 012737 000012 001122          MOV  #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3323 013126 060037 001122          ADD  RO,$BDADR     ;ADD THE I/O BASE ADDRESS
3324 013132 012737 011700 001124          MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
```

K05

MD-11-DZRIE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 62
 DZRJEA.CMB 02-NOV-76 18:40 T7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

3325	013140	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
3326	013146	016037	000012	001170		MOV	RPDS1(RO), \$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
3327	013154	013737	001170	001164		MOV	\$TMP2, \$TMP0	;COPY IT INTO 'TMP0'
3328	013162	042737	100100	001164		BIC	#ATA!VV, \$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3329	013170	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
3330	013176	016037	000012	001172		MOV	RPDS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
3331	013204	013737	001172	001166		MOV	\$TMP3, \$TMP1	;COPY IT INTO 'TMP1'
3332	013212	042737	100100	001166		BIC	#ATA!VV, \$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3333	013220	023737	001164	001166		CMP	\$TMP0, \$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3334	013226	001006				BNE	66\$;BR IF NOT
3335	013230	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3336	013234	001037				BNE	68\$;BR IF NOT
3337	013236	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3338	013240	000137	013424			JMP	70\$;BYPASS THE REST OF THE CHECKS
3339	013244	013737	001170	001126	66\$:	MOV	\$TMP2, \$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3340	013252	013737	001226	001234		MOV	PORTB, PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3341	013260	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
3342	013266	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
3343	013272	001414				BEQ	67\$;BR IF ZERO
3344	013274	013737	001224	001234		MOV	PORTA, PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3345	013302	013737	001172	001126		MOV	\$TMP3, \$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3346	013310	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
3347	013316	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
3348	013322	001004				BNE	68\$;BR IF NOT
3349	013324	012737	177777	001250	67\$:	MOV	#-1, RELERR	;SET 'RELEASE ERROR' INDICATOR
3350	013332	104026				ERROR	26	;TYPE ERROR MESSAGE 26
3351	013334	013737	001170	001126	68\$:	MOV	\$TMP2, \$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3352	013342	013737	001224	001234		MOV	PORTA, PTNBR	;CHANGE PORT NUMBER
3353	013350	042737	100000	001170		BIC	#ATA, \$TMP2	;DON'T CHECK THE ATTN BIT
3354	013356	023737	001124	001170		CMP	\$GDDAT, \$TMP2	;ALL BITS OK ?
3355	013364	001401				BEQ	69\$;BR IF OK FROM PORT A.
3356	013366	104007				ERROR	7	;REPORT ERROR
3357	013370	013737	001172	001126	69\$:	MOV	\$TMP3, \$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3358	013376	013737	001226	001234		MOV	PORTB, PTNBR	;CHANGE PORT NUMBER
3359	013404	042737	100000	001172		BIC	#ATA, \$TMP3	;DON'T CHECK THE ATTN BIT
3360	013412	023737	001124	001172		CMP	\$GDDAT, \$TMP3	;SEE IF READ OK FROM PORT B.
3361	013420	001401				BEQ	70\$;BR IF OK
3362	013422	104007				ERROR	7	;REPORT ERROR
3363	013424	000240			70\$:	NOP		
3364	013426	005737	001250			TST	RELERR	;DID DRIVE RETURN TO NEUTRAL ?
3365	013432	001402				BEQ	+.6	;BR IF IN NEUTRAL
3366	013434	000137	013710			JMP	1\$;GO WAIT FOR DRIVE TO TIMEOUT
3367	013440	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B
3368	013446	013737	001226	001234		MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3369	013454	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3370	013460	016037	000012	001126		MOV	RPDS1(RO), \$BDDAT	;GET CONTENTS OF RPDS1
3371	013466	012737	000012	001122		MOV	#RPDS1, \$BDDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3372	013474	060037	001122			ADD	RO, \$BDDADR	;ADD RH11 BASE ADDRESS
3373	013500	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
3374	013504	013737	001126	001164		MOV	\$BDDAT, \$TMP0	;MOVE REGISTER CONTENTS TO 'TMP0'
3375	013512	042737	077777	001164		BIC	#!CATA, \$TMP0	;SAVE SPECIFIED BITS
3376	013520	023737	001124	001164		CMP	\$GDDAT, \$TMP0	;COMPARE THE BITS
3377	013526	001414				BEQ	71\$;BR IF OK
3378	013530	013737	001126	001174		MOV	\$BDDAT, \$TMP4	;COPY 'BAD DATA'
3379	013536	042737	100000	001174		BIC	#ATA, \$TMP4	;CLEAR THE MASKED BITS
3380	013544	053737	001174	001124		BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT

L05

```

3381 013552 104017          ERROR 17          ;TYPE MESSAGE 17
3382 013554 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
3383 013560 000240          71$:  NOP
3384 013562 113760 001224 000010  MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
3385 013570 013737 001224 001234  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3386 013576 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
3387 013602 016037 000012 001126  MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3388 013610 012737 000012 001122  MOV     #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3389 013616 060037 001122          ADD     RO, $B0ADR      ;ADD RHI1 BASE ADDRESS
3390 013622 005037 001124          CLR     $GDDAT         ;WHAT REGISTER SHOULD BE
3391 013626 013737 001126 001164  MOV     $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
3392 013634 042737 077777 001164  BIC     #1CATA, $TMP0   ;SAVE SPECIFIED BITS
3393 013642 023737 001124 001164  CMP     $GDDAT, $TMP0   ;COMPARE THE BITS
3394 013650 001414          BEQ     73$           ;BR IF OK
3395 013652 013737 001126 001174  MOV     $BDDAT, $TMP4   ;COPY 'BAD DATA'
3396 013660 042737 100000 001174  BIC     #ATA, $TMP4     ;CLEAR THE MASKED BITS
3397 013666 053737 001174 001124  BIS     $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
3398 013674 104017          ERROR 17          ;TYPE MESSAGE 17
3399 013676 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
3400 013702 000240          73$:  NOP
3401 013704 000137 013742    JMP     2$           ;GO CHECK FOR LOOP ON ERROR
3402
3403 ;*****
3404 ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3405 ;TO RELEASE THE DRIVE
3406
3407 013710          1$:
3408 013710 113760 001224 000010  MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
3409 013716 013737 001224 001234  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3410 013724 005760 000012          TST     RPDS1(RO)      ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3411 013730 001004          BNE     2$           ;BR WHEN DRIVE RELEASED
3412 013732 005737 001254          TST     WATCH         ;CHECK THE WATCH
3413 013736 001364          BNE     1$           ;BR IF NOT ZERO
3414 013740 104036          ERROR 36          ;NO TIMEOUT WITHIN 2 SECONDS
3415 013742 000004          2$:  SCOPE          ;LOOP ?
3416
3417
3418 ;*****
3419 ;*TEST 10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
3420 ;*
3421 ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
3422 ;*
3423 ;* A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
3424 ;* NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
3425 ;*
3426 ;*****
3427 ;$T10:
3428 013744 005737 001274          TST     KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
3429 013750 001406          BEQ     2$           ;BR IF NOT
3430 013752 100002          BPL     1$           ;BR IF JUST ENTERED TEST
3431 013754 000137 002602          JMP     EXEC          ;RETURN & GET NEXT TEST NUMBER
3432 013760 012737 177777 001274  1$:  MOV     #-1, KYBCTL   ;SET SINGLE TEST INDICATOR
3433 013766 112737 000010 001102  2$:  MOVB    #10, $TSTNM   ;TEST NUMBER
3434 013774 012737 014016 001106  MOV     #TEST10, $LPADR ;LOAD LOOP ON TEST ADDRESS
3435 014002 012737 014016 001110  MOV     #TEST10, $LPERR ;LOAD LOOP ON ERROR ADDRESS
3436 014010 012737 000144 001176  MOV     #100., $TIMES  ;DO 100. ITERATIONS

```

M05

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 64
DZRJEA.CMB 02-NOV-76 18:40 T10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL

3437	014016	012706	001100		TEST10: MOV	#STACK, SP	;LOAD THE STACK POINTER
3438	014022	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
3439	014030	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3440	014036	013737	001224	001236	MOV	PORTA,SEIZPT	;ADDR OF PORT WHICH WILL ISSUE RELEASE
3441							
3442							
3443							
3444							
3445	014044	012760	000013	000000	MOV	#13,RPCS1(RO)	;ISSUE A RELEASE COMMAND
3446							
3447							
3448							
3449							
3450							
3451							
3452							
3453	014052	005037	001250		CLR	RELERR	;CLEAR THE 'RELEASE ERROR ' INDICATOR
3454	014056	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3455	014064	060037	001122		ADD	RO,\$BDADR	;ADD THE I/O BASE ADDRESS
3456	014070	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
3457	014076	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3458	014104	016037	000012	001170	MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
3459	014112	013737	001170	001164	MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
3460	014120	042737	100100	001164	BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3461	014126	113760	001226	000010	MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3462	014134	016037	000012	001172	MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
3463	014142	013737	001172	001166	MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
3464	014150	042737	100100	001166	BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3465	014156	023737	001164	001166	CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3466	014164	001006			BNE	64\$;BR IF NOT
3467	014166	005737	001164		TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3468	014172	001037			BNE	66\$;BR IF NOT
3469	014174	104046			ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3470	014176	000137	014362		JMP	68\$;BYPASS THE REST OF THE CHECKS
3471	014202	013737	001170	001126	64\$: MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3472	014210	013737	001226	001234	MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3473	014216	113760	001226	000010	MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3474	014224	005737	001164		TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
3475	014230	001414			BEQ	65\$;BR IF ZERO
3476	014232	013737	001224	001234	MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3477	014240	013737	001172	001126	MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3478	014246	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3479	014254	005737	001166		TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
3480	014260	001004			BNE	66\$;BR IF NOT
3481	014262	012737	177777	001250	65\$: MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
3482	014270	104030			ERROR	30	;TYPE ERROR MESSAGE 30
3483	014272	013737	001170	001126	66\$: MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3484	014300	013737	001224	001234	MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
3485	014306	042737	100000	001170	BIC	#ATA,\$TMP2	;DON'T CHECK THE ATTN BIT
3486	014314	023737	001124	001170	CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
3487	014322	001401			BEQ	67\$;BR IF OK FROM PORT A.
3488	014324	104007			ERROR	7	;REPORT ERROR
3489	014326	013737	001172	001126	67\$: MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3490	014334	013737	001226	001234	MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
3491	014342	042737	100000	001172	BIC	#ATA,\$TMP3	;DON'T CHECK THE ATTN BIT
3492	014350	023737	001124	001172	CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.

3493 014356 001401
3494 014360 104007
3495 014362 000240
3496 014364 000004
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506

BEQ 68\$;BR IF OK
ERROR 7 ;REPORT ERROR
38\$: NOP
SCOPE ;LOOP ?

*TEST 11 TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
*
*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.
*

3507 014366
3508 014366 005737 001274
3509 014372 001406
3510 014374 100002
3511 014376 000137 002602
3512 014402 012737 177777 001274
3513 014410 112737 000011 001102
3514 014416 012737 014440 001106
3515 014424 012737 014440 001110
3516 014432 012737 000144 001176
3517 014440 012706 001100
3518 014444 113760 001226 000010
3519 014452 013737 001226 001234
3520 014460 013737 001226 001236

TEST11:
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 #11,\$STSTNM ;TEST NUMBER
MOV #TEST11,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST11,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #100,\$TIMES ;DO 100. ITERATIONS
TEST11: MOV #STACK,SP ;LOAD THE STACK POINTER
MOVB PORTB,RPCS2(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

3525 014466 012760 000013 000000

MOV #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

3531
3532

;VERIFY THAT THE DRIVE IS IN NEUTRAL

3533 014474 005037 001250
3534 014500 012737 000012 001122
3535 014506 060037 001122
3536 014512 012737 011700 001124
3537 014520 113760 001224 000010
3538 014526 016037 000012 001170
3539 014534 013737 001170 001164
3540 014542 042737 100100 001164
3541 014550 113760 001226 000010
3542 014556 016037 000012 001172
3543 014564 013737 001172 001166
3544 014572 042737 100100 001166
3545 014600 023737 001164 001166
3546 014606 001006
3547 014610 005737 001164
3548 014614 001037

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1,\$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,\$BDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
MOV RPDS1(RO),\$T.P2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
MOV RPDS1(RO),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 64\$;BR IF NOT
TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 66\$;BR IF NOT

B06

```

3549 014616 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3550 014620 000137 015004    JMP      68$      ;BYPASS THE REST OF THE CHECKS
3551 014624 013737 001170 001126 64$: MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3552 014632 013737 001226 001234    MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3553 014640 113760 001226 000010    MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
3554 014646 005737 001164    TST    $TMP0     ;SEE IF STATUS EQ C FROM PCRT A.
3555 014652 001414    BEQ    65$      ;BR IF ZERO
3556 014654 013737 001224 001234    MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3557 014662 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3558 014670 113760 001224 000010    MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
3559 014676 005737 001166    TST    $TMP1     ;SEE IF STATUS EQ ZERO FROM PORT B.
3560 014702 001004    BNE    66$      ;BR IF NOT
3561 014704 012737 177777 001250 65$: MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3562 014712 104030    ERROR 30        ;TYPE ERROR MESSAGE 30
3563 014714 013737 001170 001126 66$: MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3564 014722 013737 001224 001234    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
3565 014730 042737 100000 001170    BIC    #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
3566 014736 023737 001124 001170    CMP    $GDDAT,$TMP2 ;ALL BITS OK ?
3567 014744 001401    BEQ    67$      ;BR IF OK FROM PORT A.
3568 014746 104007    ERROR 7         ;REPORT ERROR
3569 014750 013737 001172 001126 67$: MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3570 014756 013737 001226 001234    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
3571 014764 042737 100000 001172    BIC    #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
3572 014772 023737 001124 001172    CMP    $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3573 015000 001401    BEQ    68$      ;BR IF OK
3574 015002 104007    ERROR 7         ;REPORT ERROR
3575 015004 000240    NOP
3576 015006 000004    SCOPE          ;LOOP ?

```

```

*****
*TEST 12      TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
*
*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
* PCRT TO RELEASE THE DRIVE.
*
* A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 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 RH11 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.
*****

```

```

3598 015010          *ST12:
3599 015010 005737 001274    TST    KYBCTL    ;PERFORMING ONLY SINGLE TESTS ?
3600 015014 001406    BEQ    2$      ;BR IF NOT
3601 015016 100002    BPL    1$      ;BR IF JUST ENTERED TEST
3602 015020 000137 002602    JMP    EXEC     ;RETURN & GET NEXT TEST NUMBER
3603 015024 012737 177777 001274 1$: MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3604 015032 112737 000012 001102 2$: MOVB   #12,$STSTNM ;TEST NUMBER

```

C06

MD-11-CZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY!! 27(1006) 02-NOV-76 18:42 PAGE 67
 CZRJE.CMB 02-NOV-76 18:40 T12 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'

3605	015040	012737	015062	001106		MOV	#TEST12,\$LPADR	;LOAD LOOP ON TEST ADDRESS
3606	015046	012737	015062	001110		MOV	#TEST12,\$LPERR	;LOAD LOOP ON ERROR ADDRESS
3607	015054	012737	007640	001176		MOV	#4000,\$TIMES	;DO 4000. ITERATIONS
3608	015062	012706	001100		TEST12:	MOV	#STACK,\$P	;LOAD THE STACK POINTER
3609								
3610								::*****
3611								
3612								
3613								;SEIZE THE DRIVE THROUGH PORT A
3614	015066	113760	001224	000010		MOV	PORTA,\$RPCS2(\$RO)	;SELECT PORT A
3615	015074	013737	001224	001236		MOV	PORTA,\$SEIZPT	;STORE SEIZING PORT'S ADDRESS
3616	015102	005060	000012			CLR	\$RPS1(\$RO)	;WRITE RPS1
3617	015106	113760	001226	000010		MOV	PORTB,\$RPCS2(\$RO)	;SELECT PORT B
3618	015114	013737	001226	001234		MOV	PORTB,\$PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3619	015122	013737	001226	001240		MOV	PORTB,\$OPPR	; 'OPPOSITE' PORT ADDRESS
3620	015130	016037	000012	001126		MOV	\$RPS1(\$RO), \$SDDAT	;SEE IF DRIVE SEIZED BY PORT A
3621	015136	010037	001122			MOV	\$RO, \$SDDADR	;RH11 BASE ADDRESS
3622	015142	062737	000012	001122		ADD	#RPS1, \$SDDADR	;GENERATE BAD REGISTER ADDRESS
3623	015150	005037	001124			CLR	\$GDDAT	;REGISTER SHOULD BE ZERO
3624	015154	023737	001124	001126		CMP	\$GDDAT, \$SDDAT	;IS THE REGISTER ZERO
3625	015162	001403				BEQ	64\$;BR IF IT IS
3626	015164	104004				ERROR	4	;REPORT THE ERROR
3627	015166	000137	016364			JMP	1\$;BYPASS REST OF THE SUBTEST
3628	015172				64\$:			
3629	015172	113760	001224	000010		MOV	PORTA,\$RPCS2(\$RO)	;SELECT PORT A
3630	015200	013737	001224	001234		MOV	PORTA,\$PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3631	015206	016037	000012	001126		MOV	\$RPS1(\$RO), \$SDDAT	;SEE IF SEIZING PORT SEES CORRECT STATUS
3632	015214	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;EXPECTED STATUS
3633	015222	013737	001124	001166		MOV	\$GDDAT, \$TMP1	;USE GOOD DATA AS A MASK
3634	015230	005137	001166			COM	\$TMP1	;COMPLEMENT THE EXPECTED STATUS
3635	015234	013737	001126	001164		MOV	\$SDDAT, \$TMP0	;SAVE THE ACTUAL STATUS
3636	015242	043737	001166	001164		BIC	\$TMP1, \$TMP0	;CLEAR UNWANTED BITS
3637	015250	023737	001124	001164		CMP	\$GDDAT, \$TMP0	;ARE THE EXPECTED STATUS BITS SET ?
3638	015256	001401				BEQ	65\$;BR IF THEY ARE
3639	015260	104005				ERROR	5	;REPORT THE ERROR
3640	015262	000240			65\$:			
3641								
3642								::*****
3643								;DRIVE CLEAR THROUGH PORT A FIRST
3644								
3645	015264	012760	000011	000000		MOV	#11,\$RPS1(\$RO)	;ISSUE DRIVE CLEAR THROUGH PORT A
3646								
3647								::*****
3648								;VERIFY THAT DRIVE STILL SEIZED BY PORT A
3649								
3650	015272	113760	001226	000010		MOV	PORTB,\$RPCS2(\$RO)	;SELECT PORT B
3651	015300	013737	001226	001234		MOV	PORTB,\$PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3652	015306	005037	001244			CLR	\$CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3653	015312	016037	000012	001126		MOV	\$RPS1(\$RO), \$SDDAT	;GET CONTENTS OF RPS1
3654	015320	012737	000012	001122		MOV	#RPS1, \$SDDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3655	015326	060737	001122			ADD	\$RO, \$SDDADR	;ADD RH11 BASE ADDRESS
3656	015332	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
3657	015336	013737	001126	001164		MOV	\$SDDAT, \$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
3658	015344	042737	100000	001164		BIC	#1C7777, \$TMP0	;SAVE SPECIFIED BITS
3659	015352	023737	001124	001164		CMP	\$GDDAT, \$TMP0	;COMPARE THE BITS
3660	015360	001414				BEQ	66\$;BR IF OK

D06

```

3661 015362 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
3662 015370 042737 077777 001174      BIC      #77777,$TMP4      ;CLEAR THE MASKED BITS
3663 015376 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
3664 015404 104033      ERROR    33                ;TYPE MESSAGE 33
3665 015406 005137 001244      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
3666 015412 000240      NOP
3667 015414 113760 001224 000010      MOVB     PORTA,RPCS2(RO)    ;SELECT PORT A
3668 015422 013737 001224 001234      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3669 015430 005037 001244      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
3670 015434 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3671 015442 012737 000012 001122      MOV      #RPDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3672 015450 060037 001122      ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
3673 015454 012737 011700 001124      MOVL    #M0L!PGM!DPR!DRY!VW,$GDDAT ;WHAT REGISTER SHOULD BE
3674 015462 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
3675 015470 042737 100000 001164      BIC      #1C77777,$TMP0   ;SAVE SPECIFIED BITS
3676 015476 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
3677 015504 001414      BEQ      68$              ;BR IF OK
3678 015506 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
3679 015514 042737 077777 001174      BIC      #77777,$TMP4     ;CLEAR THE MASKED BITS
3680 015522 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
3681 015530 104033      ERROR    33                ;TYPE MESSAGE 33
3682 015532 005137 001244      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
3683 015536 000240      NOP
3684
3685 ;:*****
3686 ;NOW ISSUE MASSBUS INIT
3687
3688 015540 012760 000040 000010      MOV      #CLR,RPCS2(RO)   ;ISSUE MASSBUS INIT
3689
3690 ;:*****
3691 ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
3692
3693 015546 113760 001226 000010      MOVB     PORTB,RPCS2(RO)  ;SELECT PORT B
3694 015554 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3695 015562 005037 001244      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
3696 015566 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3697 015574 012737 000012 001122      MOV      #RPDS1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3698 015602 060037 001122      ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
3699 015606 005037 001124      CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
3700 015612 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
3701 015620 042737 100000 001164      BIC      #1C77777,$TMP0   ;SAVE SPECIFIED BITS
3702 015626 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
3703 015634 001414      BEQ      70$              ;BR IF OK
3704 015636 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
3705 015644 042737 077777 001174      BIC      #77777,$TMP4     ;CLEAR THE MASKED BITS
3706 015652 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
3707 015660 104034      ERROR    34                ;TYPE MESSAGE 34
3708 015662 005137 001244      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
3709 015666 000240      NOP
3710 015670 113760 001224 000010      MOVB     PORTA,RPCS2(RO)  ;SELECT PORT A
3711 015676 013737 001224 001234      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3712 015704 005037 001244      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
3713 015710 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3714 015716 012737 000012 001122      MOV      #RPDS1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3715 015724 060037 001122      ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
3716 015730 012737 011700 001124      MOVL    #M0L!PGM!DPR!DRY!VW,$GDDAT ;WHAT REGISTER SHOULD BE

```

E06

MC-11-DZJJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY!! 27(1006) 02-NOV-76 18:42 PAGE 69
 DZJJE.A.CMB 02-NOV-76 18:40 T12 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'

3717	015736	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
3718	015744	042737	100000	001164	BIC	#1C7777,\$TMP0	;SAVE SPECIFIED BITS
3719	015752	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
3720	015760	001414			BEQ	72\$;BR IF OK
3721	015762	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
3722	015770	042737	077777	001174	BIC	#77777,\$TMP4	;CLEAR THE MASKED BITS
3723	015776	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3724	016004	104034			ERROR	34	;TYPE MESSAGE 34
3725	016006	005137	001244		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3726	016012	000240			72\$:	NOP	
3727							
3728							;RELEASE THE DRIVE FROM PORT A
3729							
3730	016014	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
3731	016022	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3732	016030	012760	000013	000000	MOV	#13,RPCS1(RO)	;ISSUE RELEASE THROUGH PORT A
3733							
3734							;VERIFY THAT THE DRIVE IS IN NEUTRAL
3735							
3736	016036	005037	001250		CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
3737	016042	012737	000012	001122	MOV	#RPDS1,\$BDDAT	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3738	016050	060037	001122		ADD	RO,\$BDDAT	;ADD THE I/O BASE ADDRESS
3739	016054	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
3740	016062	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3741	016070	016037	000012	001170	MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
3742	016076	013737	001170	001164	MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
3743	016104	042737	100100	001164	BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3744	016112	113760	001226	000010	MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3745	016120	016037	000012	001172	MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
3746	016126	013737	001172	001166	MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
3747	016134	042737	100100	001166	BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3748	016142	023737	001164	001166	CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3749	016150	001006			BNE	74\$;BR IF NOT
3750	016152	005737	001164		TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3751	016156	001045			BNE	76\$;BR IF NOT
3752	016160	104046			ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3753	016162	000137	016362		JMP	78\$;BYPASS THE REST OF THE CHECKS
3754	016166	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT
3755	016174	013737	001226	001234	MOV	PORTB,PTNBR	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3756	016202	113760	001226	000010	MOVB	PORTB,RPCS2(RO)	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3757	016210	005737	001164		TST	\$TMP0	;SELECT PORT B.
3758	016214	001414			BEQ	75\$;SEE IF STATUS EQ 0 FROM PORT A.
3759	016216	013737	001224	001234	MOV	PORTA,PTNBR	;BR IF ZERO
3760	016224	013737	001172	001126	MOV	\$TMP3,\$BDDAT	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3761	016232	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	; 'BAD DATA' FOR ERROR TYPE OUT
3762	016240	005737	001166		TST	\$TMP1	;SELECT PORT A.
3763	016244	001012			BNE	76\$;SEE IF STATUS EQ ZERO FROM PORT B.
3764	016246	012737	177777	001250	75\$:	MOV	#-1,RELERR
3765	016254	012760	000011	000000	MOV	#11,RPCS1(RO)	;SET 'RELEASE ERROR' INDICATOR
3766	016262	012760	000013	000000	MOV	#13,RPCS1(RO)	;CLEAR THE DRIVE
3767	016270	104026			ERROR	26	;RELEASE THE DRIVE
3768	016272	013737	001170	001126	76\$:	MOV	\$TMP2,\$BDDAT
3769	016300	013737	001224	001234	MOV	PORTA,PTNBR	;TYPE ERROR MESSAGE 26
3770	016306	042737	100000	001170	BIC	#ATA,\$TMP2	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3771	016314	023737	001124	001170	CMP	\$GDDAT,\$TMP2	;CHANGE PORT NUMBER
3772	016322	001401			BEQ	77\$;DON'T CHECK THE ATTN BIT
							;ALL BITS OK ?
							;BR IF OK FROM PORT A.

F06

MD-11-DJRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY:1 27(1006) 02-NOV-76 18:42 PAGE 70
 DJRJE.CMB 02-NOV-76 18:40 T12 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'

```

3773 016324 104007          ERROR 7          ;REPORT ERROR
3774 016326 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3775 016334 013737 001226 001234      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
3776 016342 042737 100000 001172      BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
3777 016350 023737 001124 001172      CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3778 016356 001401          BEQ 78$          ;BR IF OK
3779 016360 104007          ERROR 7          ;REPORT ERROR
3780 016362 000240          78$: NOP
3781 016364 000004          1$: SCOPE          ;LOOP ?
  
```

```

3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
  
```

```

*****
*TEST 13      TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
*
*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 RPDS1 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 RH11 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.
*
*****
  
```

```

3802 016366          TST13:
3803 016366 005737 001274      TST KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3804 016372 001406          BEQ 2$             ;BR IF NOT
3805 016374 100002          BPL 1$            ;BR IF JUST ENTERED TEST
3806 016376 000137 002602      JMP EXEC          ;RETURN & GET NEXT TEST NUMBER
3807 016402 012737 177777 001274 1$: MOV #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
3808 016410 112737 000013 001102 2$: MOV #13,$TSTNM ;TEST NUMBER
3809 016416 012737 016440 001106      MOV #TEST13,$LPADR ;LOAD LOOP ON TEST ADDRESS
3810 016424 012737 016440 001110      MOV #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3811 016432 012737 007640 001176      MOV #4000,$TIMES  ;DO 4000. ITERATIONS
3812 016440 012706 001100      TEST13: MOV #STACK,SP ;LOAD THE STACK POINTER
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

3818 016444 113760 001226 000010      MOV# PORTB,RPCS2(RO) ;SELECT PORT B
3819 016452 013737 001226 001236      MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
3820 016460 005060 000012          CLR RPDS1(RO)      ;WRITE RPDS1
3821 016464 113760 001224 000010      MOV# PORTA,RPCS2(RO) ;SELECT PORT A
3822 016472 013737 001224 001234      MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3823 016500 013737 001224 001240      MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
3824 016506 016037 000012 001126      MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
3825 016514 010037 001122          MOV RO,$BDADR     ;RH11 BASE ADDRESS
3826 016520 062737 000012 001122      ADD #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
3827 016526 005037 001124          CLR $GDDAT        ;REGISTER SHOULD BE ZERO
3828 016532 023737 001124 001126      CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
  
```


G06

NO-11-DJRJE-A, DUAL CONTR. R LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 71
 DJRJE.CMB 02-NOV-76 18:43 T13 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'

```

3829 016540 001403      BEQ      64$      ;BR IF IT IS
3830 016542 104004      ERROR   4        ;REPORT THE ERROR
3831 016544 000137 017742      JMP      1$      ;BYPASS REST OF THE SUBTEST
3832 016550          64$:
3833 016550 113760 001226 000010      MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
3834 016556 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3835 016564 016037 000012 001126      MOV     RPDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
3836 016572 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
3837 016600 013737 001124 001166      MOV     $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
3838 016606 005137 001166          COM     $TMP1 ;COMPLEMENT THE EXPECTED STATUS
3839 016612 013737 001126 001164      MOV     $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
3840 016620 043737 001166 001164      BIC     $TMP1, $TMP0 ;CLEAR UNWANTED BITS
3841 016626 023737 001124 001164      CMP     $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
3842 016634 001401          BEQ     65$      ;BR IF THEY ARE
3843 016636 104005      ERROR   5        ;REPORT THE ERROR
3844 016640 000240          65$:
3845          NOP
3846          ;*****
3847          ;DRIVE CLEAR THROUGH PORT B FIRST
3848
3849 016642 012760 000011 000000      MOV     #11,RPCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT B
3850
3851          ;*****
3852          ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
3853
3854 016650 113760 001224 000010      MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
3855 016656 013737 001224 001234      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3856 016664 005037 001244          CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3857 016670 016037 000012 001126      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3858 016676 012737 000012 001122      MOV     #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3859 016704 060037 001122          ADD     RO, $B0ADR ;ADD RH11 BASE ADDRESS
3860 016710 005037 001124          CLR     $GDDAT ;WHAT REGISTER SHOULD BE
3861 016714 013737 001126 001164      MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3862 016722 042737 100000 001164      BIC     #1C7777, $TMP0 ;SAVE SPECIFIED BITS
3863 016730 023737 001124 001164      CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
3864 016736 001414          BEQ     66$      ;BR IF OK
3865 016740 013737 001126 001174      MOV     $BDDAT, $TMP4 ;COPY 'BAD DATA'
3866 016746 042737 077777 001174      BIC     #77777, $TMP4 ;CLEAR THE MASKED BITS
3867 016754 053737 001174 001124      BIS     $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3868 016762 104033      ERROR   33       ;TYPE MESSAGE 33
3869 016764 005137 001244          COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3870 016770 000240          66$:
3871 016772 113760 001226 000010      MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
3872 017000 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3873 017006 005037 001244          CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3874 017012 016037 000012 001126      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3875 017020 012737 000012 001122      MOV     #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3876 017026 060037 001122          ADD     RO, $B0ADR ;ADD RH11 BASE ADDRESS
3877 017032 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
3878 017040 013737 001126 001164      MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3879 017046 042737 100000 001164      BIC     #1C7777, $TMP0 ;SAVE SPECIFIED BITS
3880 017054 023737 001124 001164      CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
3881 017062 001414          BEQ     68$      ;BR IF OK
3882 017064 013737 001126 001174      MOV     $BDDAT, $TMP4 ;COPY 'BAD DATA'
3883 017072 042737 077777 001174      BIC     #77777, $TMP4 ;CLEAR THE MASKED BITS
3884 017100 053737 001174 001124      BIS     $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
  
```

H06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 72
DZRJEA.CMB 02-NOV-76 18:40 T13 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'

```
3885 017106 104033          ERROR 33          ;TYPE MESSAGE 33
3886 017110 005137 001244    COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
3887 017114 000240          68$:  NOP
3888
3889 ;:*****
3890 ;NOW ISSUE MASSBUS INIT
3891
3892 017116 012760 000040 000010    MOV    #CLR,RPCS2(R0) ;ISSUE MASSBUS INIT
3893
3894 ;:*****
3895 ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
3896
3897 017124 113760 001224 000010    MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
3898 017132 013737 001224 001234    MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3899 017140 005037 001244          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
3900 017144 016037 000012 001126    MOV    RPS1(R0), $BDDAT ;GET CONTENTS OF RPS1
3901 017152 012737 000012 001122    MOV    #RPS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3902 017160 060037 001122          ADD    R0, $B0ADR    ;ADD RHI1 BASE ADDRESS
3903 017164 005037 001124          CLR    $GDDAT       ;WHAT REGISTER SHOULD BE
3904 017170 013737 001126 001164    MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3905 017175 042737 100000 001164    BIC    #1C7777, $TMP0 ;SAVE SPECIFIED BITS
3906 017204 023737 001124 001164    CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
3907 017212 001414          BEQ    70$          ;BR IF OK
3908 017214 013737 001126 001174    MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
3909 017222 042737 077777 001174    BIC    #77777, $TMP4 ;CLEAR THE MASKED BITS
3910 017230 053737 001174 001124    BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3911 017236 104034          ERROR 34          ;TYPE MESSAGE 34
3912 017240 005137 001244    COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
3913 017244 000240          70$:  NOP
3914 017246 113760 001226 000010    MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
3915 017254 013737 001226 001234    MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3916 017262 005037 001244          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
3917 017266 016037 000012 001126    MOV    RPS1(R0), $BDDAT ;GET CONTENTS OF RPS1
3918 017274 012737 000012 001122    MOV    #RPS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3919 017302 060037 001122          ADD    R0, $B0ADR    ;ADD RHI1 BASE ADDRESS
3920 017306 012737 011700 001124    MOV    #M0L!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
3921 017314 013737 001126 001164    MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3922 017322 042737 100000 001164    BIC    #1C7777, $TMP0 ;SAVE SPECIFIED BITS
3923 017330 023737 001124 001164    CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
3924 017336 001414          BEQ    72$          ;BR IF OK
3925 017340 013737 001126 001174    MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
3926 017346 042737 077777 001174    BIC    #77777, $TMP4 ;CLEAR THE MASKED BITS
3927 017354 053737 001174 001124    BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3928 017362 104034          ERROR 34          ;TYPE MESSAGE 34
3929 017364 005137 001244    COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
3930 017370 000240          72$:  NOP
3931
3932 ;RELEASE THE DRIVE FROM PORT B
3933
3934 017372 113760 001226 000010    MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
3935 017400 013737 001226 001234    MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3936 017406 012760 000013 000000    MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B
3937
3938 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3939
3940 017414 005037 001250          CLR    RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
```

```

3941 017420 012737 000012 001122      MOV      #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3942 017426 060037 001122      ADD      RD,$BDDADR ;ADD THE I/O BASE ADDRESS
3943 017432 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3944 017440 113760 001224 000010      MOVVB   PORTA,RPCS2(RD) ;SELECT PORT A.
3945 017446 016037 000012 001170      MOV      RPDS1(RD),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3946 017454 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
3947 017462 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3948 017470 113760 001226 000010      MOVVB   PORTB,RPCS2(RD) ;SELECT PORT B.
3949 017476 016037 000012 001172      MOV      RPDS1(RD),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3950 017504 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
3951 017512 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3952 017520 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3953 017526 001006      BNE      74$ ;BR IF NOT
3954 017530 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3955 017534 001045      BNE      76$ ;BR IF NOT
3956 017536 104046      ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3957 017540 000137 017740      JMP      78$ ;BYPASS THE REST OF THE CHECKS
3958 017544 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3959 017552 013737 001226 001234      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3960 017560 113760 001226 000010      MOVVB   PORTB,RPCS2(RD) ;SELECT PORT B.
3961 017566 005737 001164      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3962 017572 001414      BEQ      75$ ;BR IF ZERO
3963 017574 013737 001224 001234      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3964 017602 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3965 017610 113760 001224 000010      MOVVB   PORTA,RPCS2(RD) ;SELECT PORT A.
3966 017616 005737 001166      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3967 017622 001012      BNE      76$ ;BR IF NOT
3968 017624 012737 177777 001250 75$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3969 017632 012760 000011 000000      MOV      #11,RPCS1(RD) ;CLEAR THE DRIVE
3970 017640 012760 000013 000000      MOV      #13,RPCS1(RD) ;RELEASE THE DRIVE
3971 017646 104026      ERROR    26 ;TYPE ERROR MESSAGE 26
3972 017650 013737 001170 001126 76$:      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3973 017656 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
3974 017664 042737 100000 001170      BIC      #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
3975 017672 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
3976 017700 001401      BEQ      77$ ;BR IF OK FROM PORT A.
3977 017702 104007      ERROR    ;REPORT ERROR
3978 017704 013737 001172 001126 77$:      MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3979 017712 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
3980 017720 042737 100000 001172      BIC      #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
3981 017726 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3982 017734 001401      BEQ      78$ ;BR IF OK
3983 017736 104007      ERROR    7 ;REPORT ERROR
3984 017740 000240 78$:      NOP
3985 017742 000004 1$:      SCOPE ;LOOP ?

```

```

3986
3987
3988
3989 *****
3990 *TEST 14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
3991 *
3992 *VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
3993 * SEIZING PORT.
3994 *
3995 * A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
3996 * SET.

```

J06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 74
DZRJEA.CMB 02-NOV-76 19:40 T14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052

017744
017744 005737 001274
017750 001406
017752 100002
017754 000137 002602
017760 012737 177777 001274
017766 112737 000014 001102
017774 012737 020016 001106
020002 012737 020016 001110
020010 012737 000004 001176
020016 012706 001100

020022 113760 001224 000010
020030 012760 177777 000014
020036 005060 000014
020042 013760 001226 000010
020050 005760 000012
020054 001775
020056 012760 177777 000014
020064 005060 000014
020070 113760 001224 000010
020076 005760 000012
020102 001775

020104 113760 001224 000010
020112 013777 001224 001234
020120 005037 001244
020124 016037 000012 001126
020132 012737 007012 001122
020140 060037 001122
020144 012737 100070 001124
020152 013737 001126 001164
020160 042737 077777 001164
020166 023737 001124 001164
020174 001414
020176 013737 001126 001174
020204 042737 100000 001174
020212 053737 001174 001124
020220 104010
020222 005137 001244

* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
* C. ISSUE A MASSBUS CLEAR.
* 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.

TST14: 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 #14,\$STSTNM ;TEST NUMBER
MOV #TEST14,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST14,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,\$TIMES ;DO 4 ITERATIONS
TEST14: MOV #STACK,\$P ;LOAD THE STACK POINTER
;*****
;SET ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RPCS2(R0) ;SELECT PORT 64\$
MOV #-1,RPER1(R0) ;FORCE ERRORS
CLR RPER1(R0) ;CLEAR THE ERRORS
MOV PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
54\$: TST RPDS1(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64\$;BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RPER1(R0) ;FORCE ERRORS ON PORT 65\$
CLR RPER1(R0) ;CLEAR THE ERRORS
MOVB PORTA,RPCS2(R0) ;SELECT PORT "64\$" AGAIN
65\$: TST RPDS1(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65\$;BR IF DRIVE HASN'T TIMED OUT
;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(R0),\$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,\$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,\$BDADR ;ADD RHI1 BASE ADDRESS
MOV #ATA,\$GDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT,\$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #CATA,\$STMP0 ;SAVE SPECIFIED BITS
CMP \$GDDAT,\$STMP0 ;COMPARE THE BITS
BEQ 66\$;BR IF OK
MOV \$BDDAT,\$STMP4 ;COPY 'BAD DATA'
BIC #ATA,\$STMP4 ;CLEAR THE MASKED BITS
BIS \$STMP4,\$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

K06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 75
 DZRJEA.CMB 02-NOV-76 19:40 T14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

```

4053 020226 000240
4054 020230 005737 001244
4055 020234 001402
4056 020236 000137 021244
4057 020242 113760 001226 000010
4058 020250 013737 001226 001234
4059 020256 005037 001244
4060 020262 016037 000012 001126
4061 020270 012737 000012 001122
4062 020276 060037 001122
4063 020302 012737 100000 001124
4064 020310 013737 001126 001164
4065 020316 042737 077777 001164
4066 020324 023737 001124 001164
4067 020332 001414
4068 020334 013737 001126 001174
4069 020342 042737 100000 001174
4070 020350 053737 001174 001124
4071 020356 104010
4072 020360 005137 001244
4073 020364 000240
4074 020366 005737 001244
4075 020372 001402
4076 020374 000137 021244
4077
4078
4079
4080
4081
4082 020400 113760 001224 000010
4083 020406 013737 001224 001236
4084 020414 005060 000012
4085 020420 013737 001226 001240
4086
4087
4088
4089
4090 020426 012760 000040 000010
4091 020434 113760 001224 000010
4092
4093
4094
4095
4096 020442 005037 001244
4097 020446 016037 000012 001126
4098 020454 012737 000012 001122
4099 020462 060037 001122
4100 020466 005037 001124
4101 020472 013737 001126 001164
4102 020500 042737 077777 001164
4103 020506 023737 001124 001164
4104 020514 001414
4105 020516 013737 001126 001174
4106 020524 042737 100000 001174
4107 020532 053737 001174 001124
4108 020540 104047

66$: NOP
TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
BEQ .+6 ;BR IF IT WAS
JMP IS ;BYPASS REST OF TEST IF NOT
MOV PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, $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, $TMP0 ;COMPARE THE BITS
BEQ 68$ ;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 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

68$: NOP
TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
BEQ .+6 ;BR IF IT WAS
JMP IS ;BYPASS REST OF TEST IF NOT

;*****
;SEIZE THE DRIVE THROUGH PORT A
MOV PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS

;*****
;ISSUE MASSBUS INIT TO PORT A
MOV #CLR,RPCS2(RO) ;MASSBUS INIT
MOV PORTA,RPCS2(RO) ;SELECT PORT A AGAIN

;*****
;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, $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 #+CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 72$ ;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 47 ;TYPE MESSAGE 47

```

L06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 76
 DZRJEA.CMB 02-NOV-76 18:40 T14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

```

4109 020542 005137 001244          CUM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
4110 020546 000240          72$:    NOP
4111
4112
4113
4114
4115
4116 020550 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
4117 020556 013737 001224 001234      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4118 020564 012760 000013 000000      MOV     #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164

020572 005037 001250          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
020576 012737 000012 001122      MOV     #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
020604 060037 001122          ADD     RO,$BDADR ;ADD THE I/O BASE ADDRESS
020610 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
020616 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
020624 016037 000012 001170      MOV     RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
020632 013737 001170 001164      MOV     $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
020640 042737 100100 001164      BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
020646 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
020654 016037 000012 001172      MOV     RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
020662 013737 001172 001166      MOV     $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
020670 042737 100100 001166      BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
020676 023737 001164 001166      CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
020704 001006          BNE     74$ ;BR IF NOT
020706 005737 001164          TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
020712 001045          BNE     76$ ;BR IF NOT
020714 104046          ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
020716 000137 021116          JMP     78$ ;BYPASS THE REST OF THE CHECKS
020722 013737 001170 001126 74$:    MOV     $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
020730 013737 001226 001234      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020736 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
020744 005737 001164          TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
020750 001414          BEQ     75$ ;BR IF ZERO
020752 013737 001224 001234      MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020760 013737 001172 001126      MOV     $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
020766 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
020774 005737 001166          TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
021000 001012          BNE     76$ ;BR IF NOT
021002 012737 177777 001250 75$:    MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
021010 012760 000011 000000      MOV     #11,RPCS1(RO) ;CLEAR THE DRIVE
021016 012760 000013 000000      MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
021024 104026          ERROR   26 ;TYPE ERROR MESSAGE 26
021026 013737 001170 001126 76$:    MOV     $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
021034 013737 001224 001234      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
021042 042737 100000 001170      BIC     #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
021050 023737 001124 001170      CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
021056 001401          BEQ     77$ ;BR IF OK FROM PORT A.
021060 104007          ERROR   7 ;REPORT ERROR
021062 013737 001172 001126 77$:    MOV     $TMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
021070 013737 001226 001234      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
021076 042737 100000 001172      BIC     #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
021104 023737 001124 001172      CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
021112 001401          BEQ     78$ ;BR IF OK

```

M06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 77
DZRJEA.CMB 02-NOV-76 18:40 T14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

4165 021114 104007
4166 021116 000240
4167
4168
4169
4170
4171 021120 113760 001226 000010
4172 021126 013737 001226 001234
4173 021134 005037 001244
4174 021140 016037 000012 001126
4175 021146 012737 000012 001122
4176 021154 060037 001122
4177 021160 012737 100000 001124
4178 021166 013737 001126 001164
4179 021174 042737 077777 001164
4180 021202 023737 001124 001164
4181 021210 001414
4182 021212 013737 001126 001174
4183 021220 042737 100000 001174
4184 021226 053737 001174 001124
4185 021234 104050
4186 021236 005137 001244
4187 021242 000240
4188 021244 000004
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208 021246
4209 021246 005737 001274
4210 021252 001406
4211 021254 100002
4212 021256 000137 002602
4213 021262 012737 177777 001274
4214 021270 112737 000015 001102
4215 021276 012737 021320 001106
4216 021304 012737 021320 001110
4217 021312 012737 000004 001176
4218 021320 012706 001100
4219
4220

ERROR 7 ;REPORT ERROR
78\$: NOP
;*****
;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)
MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, \$BDADR ;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 79\$;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 50 ;TYPE MESSAGE 50
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
79\$: NOP
1\$: SCOPE ;LOOP ?

;*****
;TEST 15 TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
;*
;*VERIFY THAT A MASSBUS INITIALIZE 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 RPDS1.
;*
;* C. ISSUE A MASSBUS CLEAR.
;*
;* 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.
;*****

TST15: 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\$: MOV #15, \$STNM ;TEST NUMBER
MOV #TEST15, \$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST15, \$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4, \$TIMES ;DO 4 ITERATIONS
TEST15: MOV #STACK, SP ;LOAD THE STACK POINTER
;*****

N06

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 78
DZRJEA.CMB 02-NOV-76 18:40 T15 TEST RESET ATTENTION 'B' BY MASSBUS CLEAR

;SET ATTENTION BITS FOR BOTH PORTS

```

4221
4222
4223 021324 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT 64$
4224 021332 012760 177777 000014      MOV   #-1,RPER1(RO)  ;FORCE ERRORS
4225 021340 005060 000014              CLR   RPER1(RO)      ;CLEAR THE ERRORS
4226 021344 013760 001226 000010      MOV   PORTB,RPCS2(RO);SELECT THE OTHER PORT
4227 021352 005760 000012      64$: TST  RPDS1(RO)    ;WAIT FOR DRIVE TO TIMEOUT
4228 021356 001775              BEQ   64$           ;BR IF DRIVE HASN'T TIMED OUT
4229 021360 012760 177777 000014      MOV   #-1,RPER1(RO)  ;FORCE ERRORS ON PORT 65$
4230 021366 005060 000014              CLR   RPER1(RO)      ;CLEAR THE ERRORS
4231 021372 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT "64$" AGAIN
4232 021400 005760 000012      65$: TST  RPDS1(RO)    ;WAIT FOR DRIVE TO TIMEOUT
4233 021404 001775              BEQ   65$           ;BR IF DRIVE HASN'T TIMED OUT

```

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

```

4238 021406 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
4239 021414 013737 001226 001234      MOV   PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4240 021422 005037 001244              CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
4241 021426 016037 000012 001126      MOV   RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
4242 021434 012737 000012 001122      MOV   #RPDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4243 021442 060037 001122              ADD   RO,SBADR      ;ADD RH11 BASE ADDRESS
4244 021446 012737 100000 001124      MOV   #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
4245 021454 013737 001126 001164      MOV   SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
4246 021462 042737 077777 001164      BIC   #ICATA,$TMP0  ;SAVE SPECIFIED BITS
4247 021470 023737 001124 001164      CMP   $GDDAT,$TMP0  ;COMPARE THE BITS
4248 021476 001414              BEQ   66$           ;BR IF OK
4249 021500 013737 001126 001174      MOV   SBDDAT,$TMP4  ;COPY 'BAD DATA'
4250 021506 042737 100000 001174      BIC   #ATA,$TMP4    ;CLEAR THE MASKED BITS
4251 021514 053737 001174 001124      BIS   $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
4252 021522 104010              ERROR 10           ;REPORT THE ERROR
4253 021524 005137 001244              COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
4254 021530 000240      66$: NOP
4255 021532 005737 001244              TST  CKERR          ;WAS ATTN BIT FOR PORT B SET ?
4256 021536 001402              BEQ   .+6           ;BR IF IT WAS
4257 021540 000137 022546              JMP   1$           ;BYPASS REST OF TEST IF NOT
4258 021544 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A
4259 021552 013737 001224 001234      MOV   PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4260 021560 005037 001244              CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
4261 021564 016037 000012 001126      MOV   RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
4262 021572 012737 000012 001122      MOV   #RPDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4263 021600 060037 001122              ADD   RO,SBADR      ;ADD RH11 BASE ADDRESS
4264 021604 012737 100000 001124      MOV   #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
4265 021612 013737 001126 001164      MOV   SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
4266 021620 042737 077777 001164      BIC   #ICATA,$TMP0  ;SAVE SPECIFIED BITS
4267 021626 023737 001124 001164      CMP   $GDDAT,$TMP0  ;COMPARE THE BITS
4268 021634 001414              BEQ   68$           ;BR IF OK
4269 021636 013737 001126 001174      MOV   SBDDAT,$TMP4  ;COPY 'BAD DATA'
4270 021644 042737 100000 001174      BIC   #ATA,$TMP4    ;CLEAR THE MASKED BITS
4271 021652 053737 001174 001124      BIS   $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
4272 021660 104010              ERROR 10           ;REPORT THE ERROR
4273 021662 005137 001244              COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
4274 021666 000240      68$: NOP
4275 021670 005737 001244              TST  CKERR          ;WAS ATTN BIT FOR PORT A SET ?
4276 021674 001402              BEQ   .+6           ;BR IF IT WAS

```



```

4277 021676 000137 022546 JMP IS ;BYPASS REST OF TEST IF NOT
4278
4279
4280
4281
4282
4283
4284 021702 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
4285 021710 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
4286 021716 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
4287 021722 013737 001224 001240 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
4288
4289
4290
4291
4292 021730 012760 000040 000010 MOV #CLR,RPCS2(RO) ;MASSBUS INIT
4293 021736 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B AGAIN
4294
4295
4296
4297
4298 021744 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4299 021750 016037 000012 001126 MOV RPDS1(RO),SBDAT ;GET CONTENTS OF RPDS1
4300 021756 012737 000012 001122 MOV #RPDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4301 021764 060037 001122 ADD RO,SBDADR ;ADD RHI1 BASE ADDRESS
4302 021770 005037 001124 CLR $GODAT ;WHAT REGISTER SHOULD BE
4303 021774 013737 001126 001164 MOV SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4304 022002 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
4305 022010 023737 001124 001164 CMP $GODAT,$TMP0 ;COMPARE THE BITS
4306 022016 001414 BEQ 725 ;BR IF OK
4307 022020 013737 001126 001174 MOV SBDAT,$TMP4 ;COPY 'BAD DATA'
4308 022026 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
4309 022034 053737 001174 001124 BIS $TMP4,$GODAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4310 022042 104047 ERROR 47 ;TYPE MESSAGE 47
4311 022044 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4312 022050 000240 725: NOP
4313
4314
4315
4316
4317 022052 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
4318 022060 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4319 022066 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4320
4321
4322
4323
4324 022074 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4325 022100 012737 000012 001122 MOV #RPDS1,SBDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4326 022106 060037 001122 ADD RO,SBDADR ;ADD THE I/O BASE ADDRESS
4327 022112 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GODAT ;COMPARISON CONSTANT
4328 022120 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
4329 022126 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4330 022134 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'STMP0'
4331 022142 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4332 022150 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
4333 022156 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

C07

```

4333 022164 013737 001172 001166      MOV      STMP3,STMP1      ;COPY IT INTO 'STMP1'
4334 022172 042737 100100 001166      BIC      #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4335 022200 023737 001164 001166      CMP      STMP0,STMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4336 022206 001006                BNE      74$            ;BR IF NOT
4337 022210 005737 001164                TST      STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4338 022214 001045                BNE      76$            ;BR IF NOT
4339 022216 104046                ERROR    46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4340 022220 000137 022420                JMP      78$            ;BYPASS THE REST OF THE CHECKS
4341 022224 013737 001170 001126 74$:      MOV      STMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4342 022232 013737 001226 001234                MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4343 022240 113760 001226 000010                MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
4344 022246 005737 001164                TST      STMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
4345 022252 001414                BEQ      75$            ;BR IF ZERO
4346 022254 013737 001224 001234                MOV      PORTH,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4347 022262 013737 001172 001126                MOV      STMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4348 022270 113760 001224 000010                MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
4349 022276 005737 001166                TST      STMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
4350 022302 001012                BNE      76$            ;BR IF NOT
4351 022304 012737 177777 001250 75$:      MOV      #-1,RELEARR    ;SET 'RELEASE ERROR' INDICATOR
4352 022312 012760 000011 000000                MOV      #11,RPCS1(RO)  ;CLEAR THE DRIVE
4353 022320 012760 000013 000000                MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
4354 022326 104026                ERROR    26            ;TYPE ERROR MESSAGE 26
4355 022330 013737 001170 001126 76$:      MOV      STMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPD51 READ
4356 022336 013737 001224 001234                MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
4357 022344 042737 100000 001170                BIC      #ATA,STMP2     ;DON'T CHECK THE ATTN BIT
4358 022352 023737 001124 001170                CMP      $GDDAT,STMP2   ;ALL BITS OK ?
4359 022360 001401                BEQ      77$            ;BR IF OK FROM PORT A.
4360 022362 104007                ERROR    7             ;REPORT ERROR
4361 022364 013737 001172 001126 77$:      MOV      STMP3,$BDDAT    ;CHECK RPD51 FOR BIT FAILURES - FROM PORT B.
4362 022372 013737 001226 001234                MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
4363 022400 042737 100000 001172                BIC      #ATA,STMP3     ;DON'T CHECK THE ATTN BIT
4364 022406 023737 001124 001172                CMP      $GDDAT,STMP3   ;SEE IF READ OK FROM PORT B.
4365 022414 001401                BEQ      78$            ;BR IF OK
4366 022416 104007                ERROR    7             ;REPORT ERROR
4367 022420 000240 78$:      NOP
4368
4369
4370
4371
4372 022422 113760 001224 000010                MOV      PORTA,RPCS2(RO) ;SELECT PORT A
4373 022430 013737 001224 001234                MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4374 022436 005037 001244                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
4375 022442 016037 000012 001126                MOV      RPD51(RO), $BDDAT ;GET CONTENTS OF RPD51
4376 022450 012737 000012 001122                MOV      #RPD51,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4377 022456 060037 001122                ADD      RO,$BDDADR     ;ADD RHI1 BASE ADDRESS
4378 022462 012737 100000 001124                MOV      #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
4379 022470 013737 001126 001164                MOV      $BDDAT,STMP0   ;MOVE REGISTER CONTENTS TO 'STMP0'
4380 022476 042737 077777 001164                BIC      #ICATA,STMP0   ;SAVE SPECIFIED BITS
4381 022504 023737 001124 001164                CMP      $GDDAT,STMP0   ;COMPARE THE BITS
4382 022512 001414                BEQ      79$            ;BR IF OK
4383 022514 013737 001126 001174                MOV      $BDDAT,STMP4   ;COPY 'BAD DATA'
4384 022522 042737 100000 001174                BIC      #ATA,STMP4     ;CLEAR THE MASKED BITS
4385 022530 053737 001174 001124                BIS      STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
4386 022536 104050                ERROR    50            ;TYPE MESSAGE 50
4387 022540 005137 001244                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
4388 022544 000240 79$:      NOP

```

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

4389 022546 000004
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444

```
15: SCOPE ;LOOP ?

*****
*TEST 16 TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
*
*VERIFY THAT MASSBUS CLEAR DOES NOT RESET 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 NEITHER ATTENTION BIT HAS
* RESET.
*****
TEST16:
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,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST16: MOV #STACK,$SP ;LOAD THE STACK POINTER
*****
;SET ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RPCS2(R0) ;SELECT PORT 64$
MOV #-1,RPER1(R0) ;FORCE ERRORS
CLR RPER1(R0) ;CLEAR THE ERRORS
MOV PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
64$: TST RPOS1(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RPER1(R0) ;FORCE ERRORS ON PORT 65$
CLR RPER1(R0) ;CLEAR THE ERRORS
65$: MOVB PORTA,RPCS2(R0) ;SELECT PORT "64$" AGAIN
TST RPOS1(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT

*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR 'PEOUT'
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPOS1(R0),$BDDAT ;GET CONTENTS OF RPOS1
MOV #RPOS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$B0ADR ;ADD RHI1 BASE ADDRESS
MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
```

E07

```

4445 022764 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
4446 022772 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
4447 023000 001414 BEQ 66$ ;BR IF OK
4448 023002 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
4449 023010 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
4450 023016 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4451 023024 104010 ERROR 10 ;REPORT THE ERROR
4452 023026 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4453 023032 000240 66$: NOP
4454 023034 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
4455 023040 001402 BEQ .+6 ;BR IF IT WAS
4456 023042 000137 024006 JMP 1$ ;BYPASS REST OF TEST IF NOT
4457 023046 113760 001226 000010 MOV#B PORTB,RPCS2(R0) ;SELECT PORT B
4458 023054 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4459 023062 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4460 023066 016037 000012 001126 MOV RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
4461 023074 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4462 023102 060037 001122 ADD R0,$B0ADR ;ADD RHI1 BASE ADDRESS
4463 023106 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
4464 023114 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4465 023122 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
4466 023130 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
4467 023136 001414 BEQ 68$ ;BR IF OK
4468 023140 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
4469 023146 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
4470 023154 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4471 023162 104010 ERROR 10 ;REPORT THE ERROR
4472 023164 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4473 023170 000240 68$: NOP
4474 023172 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
4475 023176 001402 BEQ .+6 ;BR IF IT WAS
4476 023200 000137 024006 JMP 1$ ;BYPASS REST OF TEST IF NOT
4477
4478 ;*****
4479
4480 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4481
4482 023204 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4483 023210 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4484 023216 060037 001122 ADD R0,$B0ADR ;ADD THE I/O BASE ADDRESS
4485 023222 012737 111700 001124 MOV #111700,$GDDAT ;COMPARISON CONSTANT
4486 023230 113760 001224 000010 MOV#B PORTA,RPCS2(R0) ;SELECT PORT A.
4487 023236 016037 000012 001170 MOV RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4488 023244 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
4489 023252 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4490 023260 113760 001226 000010 MOV#B PORTB,RPCS2(R0) ;SELECT PORT B.
4491 023266 016037 000012 001172 MOV RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4492 023274 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
4493 023302 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4494 023310 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4495 023316 001006 BNE 70$ ;BR IF NOT
4496 023320 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4497 023324 001045 BNE 72$ ;BR IF NOT
4498 023326 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4499 023330 000137 023514 JMP 74$ ;BYPASS THE REST OF THE CHECKS
4500 023334 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
  
```

F07

4501	023342	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4502	023350	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
4503	023356	005737	001164			TST	\$TMPD	;SEE IF STATUS EQ 0 FROM PORT A.
4504	023362	001414				BEG	71\$;BR IF ZERO
4505	023364	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4506	023372	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
4507	023400	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
4508	023406	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
4509	023412	001012				BNE	72\$;BR IF NOT
4510	023414	012737	177777	001250	71\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
4511	023422	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
4512	023430	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
4513	023436	104026				ERROR	26	;TYPE ERROR MESSAGE 26
4514	023440	013737	001170	001126	72\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4515	023446	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
4516	023454	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
4517	023462	001401				BEG	73\$;BR IF OK FROM PORT A.
4518	023464	104007				ERROR	7	;REPORT ERROR
4519	023466	013737	001172	001126	73\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4520	023474	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
4521	023502	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
4522	023510	001401				BEG	74\$;BR IF OK
4523	023512	104007				ERROR	7	;REPORT ERROR
4524	023514	000240			74\$:	NOP		
4525	023516	005737	001250			TST	RELERR	;WAS DRIVE IN NEUTRAL ?
4526	023522	001402				BEG	+6	;BR IF IT WAS
4527	023524	000137	024006			JMP	1\$;BYPASS RESET OF TEST
4528						;*****		
4529						;ISSUE THE MASSBUS INIT		
4530						;*****		
4531	023530	012760	000040	000010		MOV	#CLR,RPCS2(RO)	;ISSUE A MASSBUS INIT
4532						;*****		
4533						;CHECK THE ATTENTION BITS OF BOTH PORTS		
4534						;*****		
4535						;*****		
4536	023536	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A
4537	023544	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4538	023552	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
4539	023556	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;GET CONTENTS OF RPDS1
4540	023564	012737	000012	001122		MOV	#RPDS1,\$BDDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
4541	023572	060037	001122			ADD	RO,\$BDDADR	;ADD RH11 BASE ADDRESS
4542	023576	012737	100000	001124		MOV	#ATA,\$GDDAT	;WHAT REGISTER SHOULD BE
4543	023604	013737	001126	001164		MOV	\$BDDAT,\$TMPD	;MOVE REGISTER CONTENTS TO 'TMPD'
4544	023612	042737	077777	001164		BIC	#ICATA,\$TMPD	;SAVE SPECIFIED BITS
4545	023620	023737	001124	001164		CMP	\$GDDAT,\$TMPD	;COMPARE THE BITS
4546	023626	001414				BEG	75\$;BR IF OK
4547	023630	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
4548	023636	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
4549	023644	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
4550	023652	104051				ERROR	51	;TYPE MESSAGE 51
4551	023654	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
4552	023660	000240			75\$:	NOP		
4553	023662	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B
4554	023670	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4555	023676	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
4556	023702	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;GET CONTENTS OF RPDS1

G07

```

4557 023710 012737 000012 001122      MOV      #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4558 023716 060037 001122      ADD      R0,$BDADR    ;ADD RH11 BASE ADDRESS
4559 023722 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
4560 023730 013737 001126 001164      MOV      $BDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4561 023736 042737 077777 001164      BIC      #ICATA,$TMP0 ;SAVE SPECIFIED BITS
4562 023744 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
4563 023752 001414      BEQ      77$          ;BR IF OK
4564 023754 013737 001126 001174      MOV      $BDAT,$TMP4 ;COPY 'BAD DATA'
4565 023762 042737 100000 001174      BIC      #ATA,$TMP4  ;CLEAR THE MASKED BITS
4566 023770 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4567 023776 104051      ERROR   51          ;TYPE MESSAGE 51
4568 024000 005137 001244      COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
4569 024004 000240      77$:    NOP
4570 024006 000004      1$:    SCOPE        ;LOOP ?

```

```

*****
*TEST 17      TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'
*
*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
*
*  A.  READ THE CONTROL REGISTER (RPCS1) 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.
*
*****

```

```

4585 024010      †ST17:
4586 024010 005737 001274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
4587 024014 001406      BEQ      2$          ;BR IF NOT
4588 024016 100002      BPL      1$          ;BR IF JUST ENTERED TEST
4589 024020 000137 002602      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
4590 024024 012737 177777 001274 1$:    MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
4591 024032 112737 000017 001102 2$:    MOV      #17,$STNM  ;TEST NUMBER
4592 024040 012737 024062 001106      MOV      #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
4593 024046 012737 024062 001110      MOV      #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4594 024054 012737 007640 001176      MOV      #4000,$TIMES ;DO 4000. ITERATIONS
4595 024062 012706 001100      TEST17: MOV     #STACK,$SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

4599 024066 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT #A
4600 024074 005060 000012      CLR      RPDS1(R0)      ;SEIZE THE DRIVE
4601 024100 012760 000011 000000      MOV      #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
4602 024106 012760 000013 000000      MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
4603 024114 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT #B
4604 024122 005060 000012      CLR      RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
4605 024126 012760 000011 000000      MOV      #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
4606 024134 012760 000013 000000      MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE

```

;*****

;SEIZE THE DRIVE THROUGH PORT A

```

4612 024142 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A

```

H07

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 85
 DZRJEA.CMB 02-NOV-76 18:40 T17 TEST SEIZE B' RPCS1 READ THROUGH PORT 'A'

4613	024150	013737	001224	001236		MOV	PORTA,SEIZPT	;STORE SEIZING PORT'S ADDRESS
4614	024156	005760	000000			TST	RPCS1(RO)	;READ RHCS1
4615	024162	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
4616	024170	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4617	024176	013737	001226	001240		MOV	PORTB,OPPR	; 'OPPOSITE' PORT ADDRESS
4618	024204	016037	000012	001126		MOV	RPDS1(RO), \$BDDAT	;SEE IF DRIVE SEIZED BY PORT A
4619	024212	010037	001122			MOV	RO, \$BDADR	;RH11 BASE ADDRESS
4620	024216	062737	000012	001122		ADD	#RPDS1, \$BDADR	;GENERATE BAD REGISTER ADDRESS
4621	024224	005037	001124			CLR	\$GDDAT	;REGISTER SHOULD BE ZERO
4622	024230	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	;IS THE REGISTER ZERO
4623	024236	001403				BEQ	64\$;BR IF IT IS
4624	024240	104004				ERROR	4	;REPORT THE ERROR
4625	024242	000137	024674			JMP	1\$;BYPASS REST OF THE SUBTEST
4626	024246				64\$:			
4627	024246	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
4628	024254	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4629	024262	016037	000012	001126		MOV	RPDS1(RO), \$BDDAT	;SEE IF SEIZING PORT SEES CORRECT STATUS
4630	024270	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;EXPECTED STATUS
4631	024276	013737	001124	001166		MOV	\$GDDAT, \$TMP1	;USE GOOD DATA AS A MASK
4632	024304	005137	001166			COM	\$TMP1	;COMPLEMENT THE EXPECTED STATUS
4633	024310	013737	001126	001164		MOV	\$BDDAT, \$TMP0	;SAVE THE ACTUAL STATUS
4634	024316	043737	001166	001164		BIC	\$TMP1, \$TMP0	;CLEAR UNWANTED BITS
4635	024324	023737	001124	001164		CMP	\$GDDAT, \$TMP0	;ARE THE EXPECTED STATUS BITS SET ?
4636	024332	001401				BEQ	65\$;BR IF THEY ARE
4637	024334	104005				ERROR	5	;REPORT THE ERROR
4638	024336	000240			65\$:	NOP		
4639								
4640								
4641								
4642								
4643								
4644	024340	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
4645	024346	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4646	024354	012760	000013	000000		MOV	#13, RPCS1(RO)	;ISSUE RELEASE THROUGH PORT A
4647								
4648								
4649								
4650	024362	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
4651	024366	012737	000012	001122		MOV	#RPDS1, \$BDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4652	024374	060037	001122			ADD	RO, \$BDADR	;ADD THE '0' BASE ADDRESS
4653	024400	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;COMPARISON CONSTANT
4654	024406	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
4655	024414	016037	000012	001170		MOV	RPDS1(RO), \$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
4656	024422	013737	001170	001164		MOV	\$TMP2, \$TMP0	;COPY IT INTO 'TMP0'
4657	024430	042737	100100	001164		BIC	#ATA!VV, \$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
4658	024436	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
4659	024444	016037	000012	001172		MOV	RPDS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
4660	024452	013737	001172	001166		MOV	\$TMP3, \$TMP1	;COPY IT INTO 'TMP1'
4661	024460	042737	100100	001166		BIC	#ATA!VV, \$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
4662	024466	023737	001164	001166		CMP	\$TMP0, \$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4663	024474	001006				BNE	66\$;BR IF NOT
4664	024476	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4665	024502	001045				BNE	68\$;BR IF NOT
4666	024504	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4667	024506	000137	024672			JMP	70\$;BYPASS THE REST OF THE CHECKS
4668	024512	013737	001170	001126	66\$:	MOV	\$TMP2, \$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE

;;*****

;RELEASE THE DRIVE FROM PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

4669	024520	013737	001226	001234	MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4670	024526	113760	001226	000010	MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
4671	024534	005737	001164		TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
4672	024540	001414			BEQ	67\$;BR IF ZERO
4673	024542	013737	001224	001234	MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4674	024550	013737	001172	001126	MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
4675	024556	113760	001224	000010	MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
4676	024564	005737	001166		TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
4677	024570	001012			BNE	68\$;BR IF NOT
4678	024572	012737	177777	001250	67\$: MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
4679	024600	012760	000011	000000	MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
4680	024606	012760	000013	000000	MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
4681	024614	104026			ERROR	26	;TYPE ERROR MESSAGE 26
4682	024616	013737	001170	001126	68\$: MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4683	024624	013737	001224	001234	MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
4684	024632	023737	001124	001170	CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
4685	024640	001401			BEQ	69\$;BR IF OK FROM PORT A.
4686	024642	104007			ERROR	7	;REPORT ERROR
4687	024644	013737	001172	001126	69\$: MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4688	024652	013737	001226	001234	MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
4689	024660	023737	001124	001172	CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
4690	024666	001401			BEQ	70\$;BR IF OK
4691	024670	104007			ERROR	7	;REPORT ERROR
4692	024672	000240			70\$: NOP		
4693	024674	000004			1\$: SCOPE		;LOOP ?

```

*****
;TEST 20 TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
;
;VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
;
; A. READ THE CONTROL REGISTER (RPCS1) 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.
;
*****
    
```

4707	024676				TST20:		
4708	024676	005737	001274		TST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
4709	024702	001406			BEQ	2\$;BR IF NOT
4710	024704	100002			BPL	1\$;BR IF JUST ENTERED TEST
4711	024706	000137	002602		JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
4712	024712	012737	177777	001274	1\$: MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR
4713	024720	112737	000020	001102	2\$: MOV	#20,\$STNM	;TEST NUMBER
4714	024726	012737	024750	001106	MOV	#TEST20,\$LPADR	;LOAD LOOP ON TEST ADDRESS
4715	024734	012737	024750	001110	MOV	#TEST20,\$LPERR	;LOAD LOOP ON ERROR ADDRESS
4716	024742	012737	007640	001176	MOV	#4000,\$TIMES	;DO 4000. ITERATIONS
4717	024750	012706	001100		TEST20: MOV	#STACK,SP	;LOAD THE STACK POINTER
4718							
4719							
4720							;CLEAR ATTENTION BITS FOR BOTH PORTS
4721	024754	113760	001224	000010	MOV	PORTA,RPCS2(RO)	;SELECT PORT #A
4722	024762	005060	000012		CLR	RPDS1(RO)	;SEIZE THE DRIVE
4723	024766	012760	000011	000000	MOV	#11,RPCS1(RO)	;ISSUE DRIVE CLEAR
4724	024774	012760	000013	000000	MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE

J07

4725 025002 113760 001226 000010
4726 025010 005060 000012
4727 025014 012760 000011 000000
4728 025022 012760 000013 000000
4729
4730
4731
4732
4733
4734 025030 113760 001226 000010
4735 025036 013737 001226 001236
4736 025044 005760 000000
4737 025050 113760 001224 000010
4738 025056 013737 001224 001234
4739 025064 013737 001224 001240
4740 025072 016037 000012 001126
4741 025100 010037 001122
4742 025104 062737 000012 001122
4743 025112 005037 001124
4744 025116 023737 001124 001126
4745 025124 001403
4746 025126 104004
4747 025130 000137 025562
4748 025134
4749 025134 113760 001226 000010
4750 025142 013737 001226 001234
4751 025150 016037 000012 001126
4752 025156 012737 011700 001124
4753 025164 013737 001124 001166
4754 025172 005137 001166
4755 025176 013737 001126 001164
4756 025204 043737 001166 001164
4757 025212 023737 001124 001164
4758 025220 001401
4759 025222 104005
4760 025224 000240
4761
4762
4763
4764
4765
4766 025226 113760 001226 000010
4767 025234 013737 001226 001234
4768 025242 012760 000013 000000
4769
4770
4771
4772 025250 005037 001250
4773 025254 012737 000012 001122
4774 025262 060037 001122
4775 025266 012737 011700 001124
4776 025274 113760 001224 000010
4777 025302 016037 000012 001170
4778 025310 013737 001170 001164
4779 025316 042737 100100 001164
4780 025324 113760 001226 000010

MOV B PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE

;;*****

;SEIZE THE DRIVE THROUGH PORT B

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RPCS1(RO) ;READ RHCS1
MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,SBDAADR ;RH11 BASE ADDRESS
ADD #RPDS1,SBDAADR ;GENERATE BAD REGISTER ADDRESS
CLR \$GDDAT ;REGISTER SHOULD BE ZERO
CMP \$GDDAT,SBDDAT ;IS THE REGISTER ZERO
BEQ 64\$;BR IF IT IS
ERROR 4 ;REPORT THE ERROR
JMP 1\$;BYPASS REST OF THE SUBTEST

64\$:

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
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 SBDDAT,\$TMP0 ;SAVE THE ACTUAL STATUS
BIC \$TMP1,\$TMP0 ;CLEAR UNWANTED BITS
CMP \$GDDAT,\$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
BEQ 65\$;BR IF THEY ARE
ERROR 5 ;REPORT THE ERROR

65\$:

NOB

;;*****

;RELEASE THE DRIVE FROM PORT B

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1,SBDAADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,SBDAADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
MOV B PORTA,RPCS2(RO) ;SELECT PORT A.
MOV RPDS1(RO),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOV B PORTB,RPCS2(RO) ;SELECT PORT B.

K07

```

4781 025332 016037 000012 001172      MOV      RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4782 025340 013737 001172 001166      MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
4783 025346 042737 100100 001166      BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4784 025354 023737 001164 001166      CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4785 025362 001006                BNE      66$ ;BR IF NOT
4786 025364 005737 001164                TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4787 025370 001045                BNE      68$ ;BR IF NOT
4788 025372 104046                ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4789 025374 000137 025560                JMP      70$ ;BYPASS THE REST OF THE CHECKS
4790 025400 013737 001170 001126 66$: MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4791 025406 013737 001226 001234      MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4792 025414 113760 001226 000010      MOV      PORTB, RPCS2(RO) ;SELECT PORT B.
4793 025422 005737 001164                TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4794 025426 001414                BEQ      67$ ;BR IF ZERO
4795 025430 013737 001224 001234      MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4796 025436 013737 001172 001126      MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4797 025444 113760 001224 000010      MOV      PORTA, RPCS2(RO) ;SELECT PORT A.
4798 025452 005737 001166                TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4799 025456 001012                BNE      68$ ;BR IF NOT
4800 025460 012737 177777 001250 67$: MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
4801 025466 012760 000011 000000      MOV      #11, RPCS1(RO) ;CLEAR THE DRIVE
4802 025474 012760 000013 000000      MOV      #13, RPCS1(RO) ;RELEASE THE DRIVE
4803 025502 104026                ERROR    26 ;TYPE ERROR MESSAGE 26
4804 025504 013737 001170 001126 68$: MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4805 025512 013737 001224 001234      MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
4806 025520 023737 001124 001170      CMP      $GDDAT, $TMP2 ;ALL BITS OK ?
4807 025526 001401                BEQ      69$ ;BR IF OK FROM PORT A.
4808 025530 104007                ERROR    7 ;REPORT ERROR
4809 025532 013737 001172 001126 69$: MOV      $TMP3, $BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4810 025540 013737 001226 001234      MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
4811 025546 023737 001124 001172      CMP      $GDDAT, $TMP3 ;SEE IF READ OK FROM PORT B.
4812 025554 001401                BEQ      70$ ;BR IF OK
4813 025556 104007                ERROR    7 ;REPORT ERROR
4814 025560 000240                NOP ;
4815 025562 000004                IS:     SCOPE ;LOOP ?

```

```

4816
4817
4818 *****
4819 *TEST 21 TEST 'PORT REQUEST' FROM PORT 'A'
4820 *
4821 *VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
4822 * DRIVE IS SEIZED BY THE OTHER PORT.
4823 *
4824 * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
4825 *
4826 * B. WRITE 0'S INTO RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
4827 * SEIZED BY PORT 'B'.
4828 *
4829 * C. ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
4830 * SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR
4831 * PORT 'A' AND IS NOT SET FOR PORT 'B'.
4832 *
4833 * D. ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
4834 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4835 *
4836 *****

```

```

4837 025564          TST21:
4838 025564 005737 001274      TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4839 025570 001406          BEQ    2$              ;BR IF NOT
4840 025572 100002          BPL    1$              ;BR IF JUST ENTERED TEST
4841 025574 000137 002602      JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
4842 025600 012737 177777 001274 1$:  MOV    #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4843 025606 112737 000021 001102 2$:  MOVB   #21,$STNM       ;TEST NUMBER
4844 025614 012737 025636 001106      MOV    #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS
4845 025622 012737 025636 001110      MOV    #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4846 025630 012737 007640 001176      MOV    #4000,$TIMES   ;DO 4000. ITERATIONS
4847 025636 012706 001100      TEST21: MOV    #STACK,SP   ;LOAD THE STACK POINTER
4848
4849                ;CLEAR ATTENTION BITS FOR BOTH PORTS
4850
4851 025642 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT #A
4852 025650 005060 000012          CLR    RPDS1(RO)       ;SEIZE THE DRIVE
4853 025654 012760 000011 000000      MOV    #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
4854 025662 012760 000013 000000      MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
4855 025670 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT #B
4856 025676 005060 000012          CLR    RPDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
4857 025702 012760 000011 000000      MOV    #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
4858 025710 012760 000013 000000      MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
4859
4860                ;*****
4861
4862                ;SEIZE THE DRIVE THROUGH PORT B
4863
4864 025716 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
4865 025724 013737 001226 001236      MOV    PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
4866 025732 005060 000012          CLR    RPDS1(RO)       ;WRITE RPDS1
4867 025736 013737 001224 001240      MOV    PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
4868 025744 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
4869 025752 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4870
4871                ;*****
4872                ;SET PORT REQUEST
4873
4874 025760 005060 000012          CLR    RPDS1(RO)       ;SET PORT REQUEST FOR PORT A
4875
4876                ;*****
4877                ;RELEASE THROUGH PORT B. DRIVE SHOULD SWITCH TO PORT A.
4878
4879                ;RELEASE THE DRIVE FROM PORT B
4880
4881
4882 025764 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
4883 025772 013737 001226 001234      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4884 026000 012760 000013 000000      MOV    #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT B
4885
4886                ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
4887
4888 026006 005037 001250          CLR    RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
4889 026012 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4890 026020 012737 000012 001122      MOV    #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
4891 026026 060037 001122          ADD    RO,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
4892 026032 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A

```

M07

```

4893 026040 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4894 026046 016037 000012 001164 MOV RPDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
4895 026054 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
4896 026062 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4897 026070 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
4898 026076 001404 BEQ 66$ ;BR IF STATUS FROM PORT B ZERO
4899 026100 005737 001164 TST $TMP0 ;IS STATUS FROM PORT A ZERO ?
4900 026104 001401 BEQ 66$ ;BR IF ZERO
4901 026106 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
4902 026110 013737 001164 001126 66$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
4903 026116 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
4904 026124 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
4905 026132 001401 BEQ 67$ ;BR IF OK
4906 026134 104027 ERROR 27 ;REPORT REGISTER ERROR
4907 026136 000240 67$: NOP
4908 026140 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
4909 026146 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4910 026154 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4911 026160 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
4912 026166 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4913 026174 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
4914 026200 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
4915 026204 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4916 026212 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
4917 026220 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
4918 026226 001414 BEQ 68$ ;BR IF OK
4919 026230 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
4920 026236 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
4921 026244 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4922 026252 104016 ERROR 16 ;TYPE MESSAGE 16
4923 026254 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4924 026260 000240 68$: NOP
4925 026262 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
4926 026270 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4927 026276 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4928 026302 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
4929 026310 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4930 026316 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
4931 026322 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
4932 026330 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4933 026336 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
4934 026344 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
4935 026352 001414 BEQ 70$ ;BR IF OK
4936 026354 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
4937 026362 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
4938 026370 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4939 026376 104016 ERROR 16 ;TYPE MESSAGE 16
4940 026400 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4941 026404 000240 70$: NOP
4942
4943 ;*****
4944
4945 ;RELEASE THE DRIVE FROM PORT A
4946
4947 026406 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
4948 026414 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

4949 026422 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4950
4951                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4952
4953 026430 005037 001250      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4954 026434 012737 000012 001122      MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4955 026442 060037 001122      ADD      RO,$BDADR ;ADD THE I/O BASE ADDRESS
4956 026446 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4957 026454 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
4958 026462 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4959 026470 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
4960 026476 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4961 026504 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
4962 026512 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4963 026520 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
4964 026526 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4965 026534 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4966 026542 001006                                     BNE      72$ ;BR IF NOT
4967 026544 005737 001164                                     TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4968 026550 001045                                     BNE      74$ ;BR IF NOT
4969 026552 104046                                     ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4970 026554 000137 026740                                     JMP      76$ ;BYPASS THE REST OF THE CHECKS
4971 026560 013737 001170 001126 72$:      MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4972 026566 013737 001226 001234      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4973 026574 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
4974 026602 005737 001164      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4975 026606 001414      BEQ      73$ ;BR IF ZERO
4976 026610 013737 001224 001234      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4977 026616 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4978 026624 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
4979 026632 005737 001166      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4980 026636 001012      BNE      74$ ;BR IF NOT
4981 026640 012737 177777 001250 73$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4982 026646 012760 000011 000000      MOV      #11,RPCS1(RO) ;CLEAR THE DRIVE
4983 026654 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
4984 026662 104026      ERROR    26 ;TYPE ERROR MESSAGE 26
4985 026664 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4986 026672 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
4987 026700 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
4988 026706 001401      BEQ      75$ ;BR IF OK FROM PORT A.
4989 026710 104007      ERROR    7 ;REPORT ERROR
4990 026712 013737 001172 001126 75$:      MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4991 026720 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
4992 026726 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
4993 026734 001401      BEQ      76$ ;BR IF OK
4994 026736 104007      ERROR    7 ;REPORT ERROR
4995 026740 000240      NOP
4996 026742 000004      1$:      SCOPE ;LOOP ?

```

```

4997
4998 ;*****
4999 ;*TEST 22 TEST PORT REQUEST FROM PORT 'B'
5000 ;*
5001 ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
5002 ;* DRIVE IS SEIZED BY THE OTHER PORT.
5003 ;*
5004 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

```

- * B. WRITE 0'S INTO RPDS1 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.

50000
50001
50002
50003
50004
50005
50006
50007
50008
50009
50010
50011
50012
50013
50014
50015
50016
50017
50018
50019
50020
50021
50022
50023
50024
50025
50026
50027
50028
50029
50030
50031
50032
50033
50034
50035
50036
50037
50038
50039
50040
50041
50042
50043
50044
50045
50046
50047
50048
50049
50050
50051
50052
50053
50054
50055
50056
50057
50058
50059
50060

```

*****
TST22:
      YST      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     #22,$TSTNM   ;TEST NUMBER
      MOV      #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #4000,$TIMES  ;DO 4000. ITERATIONS
TEST22: MOV     #STACK,SP   ;LOAD THE STACK POINTER

      ;CLEAR ATTENTION BITS FOR BOTH PORTS
      MOVB     PORTA,RPDS2(RO) ;SELECT PORT #A
      CLR      RPDS1(RO)      ;SEIZE THE DRIVE
      MOV      #11,RPDS1(RO)  ;ISSUE DRIVE CLEAR
      MOV      #13,RPDS1(RO)  ;RELEASE THE DRIVE
      MOVB     PORTB,RPDS2(RO) ;SELECT PORT #B
      CLR      RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
      MOV      #11,RPDS1(RO)  ;ISSUE DRIVE CLEAR
      MOV      #13,RPDS1(RO)  ;RELEASE THE DRIVE

;*****
      ;SEIZE THE DRIVE THROUGH PORT A
      MOVB     PORTA,RPDS2(RO) ;SELECT PORT A
      MOV      PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
      CLR      RPDS1(RO)      ;WRITE RPDS1
      MOV      PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
      MOVB     PORTB,RPDS2(RO) ;SELECT PORT B
      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;SET PORT REQUEST
      CLR      RPDS1(RO)      ;SET PORT REQUEST FOR PORT B

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

;RELEASE THE DRIVE FROM PORT A

```

5061						
5062	027144	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5063	027152	013737	001224	001234		MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5064	027160	012760	000013	000000		MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5065						
5066						;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5067						
5068	027166	005037	001250			CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5069	027172	012737	111700	001124		MOV #ATA:MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
5070	027200	012737	000012	001122		MOV #RPDS1,\$BDAOR ;REGISTER ADDRESS INCREMENT
5071	027206	060037	001122			ADD RO,\$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
5072	027212	113760	001226	000010		MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5073	027220	013737	001226	001234		MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5074	027226	016037	000012	001164		MOV RPDS1(RO),\$TMP0 ;READ STATUS REGISTER FROM PORT B
5075	027234	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5076	027242	013737	001224	001234		MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5077	027250	016037	000012	001126		MOV RPDS1(RO),\$BDDAT ;DRIVE STATUS FROM PORT A
5078	027256	001404				BEQ 66\$;BR IF STATUS FROM PORT A ZERO
5079	027260	005737	001164			TST \$TMP0 ;IS STATUS FROM PORT B ZERO ?
5080	027264	001401				BEQ 66\$;BR IF ZERO
5081	027266	104031				ERROR 31 ;REPORT DRIVE IN NEUTRAL
5082	027270	013737	001164	001126	66\$:	MOV \$TMP0,\$BDDAT ;CHECK STATUS FROM PORT B
5083	027276	013737	001226	001234		MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5084	027304	023737	001124	001126		CMP \$GDDAT,\$BDDAT ;COMPARE WITH CONSTANT
5085	027312	001401				BEQ 67\$;BR IF OK
5086	027314	104027				ERROR 27 ;REPORT REGISTER ERROR
5087	027316	000240			67\$:	NOP
5088	027320	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5089	027326	013737	001224	001234		MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5090	027334	005037	001244			CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5091	027340	016037	000012	001126		MOV RPDS1(RO),\$BDDAT ;GET CONTENTS OF RPDS1
5092	027346	012737	000012	001122		MOV #RPDS1,\$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5093	027354	060037	001122			ADD RO,\$BDAOR ;ADD RH11 BASE ADDRESS
5094	027360	005037	001124			CLR \$GDDAT ;WHAT REGISTER SHOULD BE
5095	027364	013737	001126	001164		MOV \$BDDAT,\$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
5096	027372	042737	077777	001164		BIC #ICATA,\$TMP0 ;SAVE SPECIFIED BITS
5097	027400	023737	001124	001164		CMP \$GDDAT,\$TMP0 ;COMPARE THE BITS
5098	027406	001414				BEQ 68\$;BR IF OK
5099	027410	013737	001126	001174		MOV \$BDDAT,\$TMP4 ;COPY 'BAD DATA'
5100	027416	042737	100000	001174		BIC #ATA,\$TMP4 ;CLEAR THE MASKED BITS
5101	027424	053737	001174	001124		BIS \$TMP4,\$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5102	027432	104016				ERROR 16 ;TYPE MESSAGE 16
5103	027434	005137	001244			COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5104	027440	000240			68\$:	NOP
5105	027442	113760	001226	000010		MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5106	027450	013737	001226	001234		MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5107	027456	005037	001244			CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5108	027462	016037	000012	001126		MOV RPDS1(RO),\$BDDAT ;GET CONTENTS OF RPDS1
5109	027470	012737	000012	001122		MOV #RPDS1,\$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5110	027476	060037	001122			ADD RO,\$BDAOR ;ADD RH11 BASE ADDRESS
5111	027502	012737	100000	001124		MOV #ATA,\$GDDAT ;WHAT REGISTER SHOULD BE
5112	027510	013737	001126	001164		MOV \$BDDAT,\$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
5113	027516	042737	077777	001164		BIC #ICATA,\$TMP0 ;SAVE SPECIFIED BITS
5114	027524	023737	001124	001164		CMP \$GDDAT,\$TMP0 ;COMPARE THE BITS
5115	027532	001414				BEQ 70\$;BR IF OK
5116	027534	013737	001126	001174		MOV \$BDDAT,\$TMP4 ;COPY 'BAD DATA'

D08

```

5117 027542 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
5118 027550 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
5119 027556 104016          ERROR    16             ;TYPE MESSAGE 16
5120 027560 005137 001244          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
5121 027564 000240          70$:    NOP
5122
5123      ;:*****
5124
5125      ;RELEASE THE DRIVE FROM PORT B
5126
5127 027566 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
5128 027574 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5129 027602 012760 000013 000000      MOV      #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT B
5130
5131      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5132
5133 027610 005037 001250          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
5134 027614 012737 000012 001122      MOV      #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5135 027622 060037 001122          ADD      RO,$BDDADR     ;ADD THE I/O BASE ADDRESS
5136 027626 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5137 027634 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
5138 027642 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5139 027650 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
5140 027656 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5141 027664 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
5142 027672 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5143 027700 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
5144 027706 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5145 027714 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5146 027722 001006          BNE      72$           ;BR IF NOT
5147 027724 005737 001164          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5148 027730 001045          BNE      74$           ;BR IF NOT
5149 027732 104046          ERROR    46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5150 027734 000137 030120          JMP      76$           ;BYPASS THE REST OF THE CHECKS
5151 027740 013737 001170 001126 72$:    MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5152 027746 013737 001226 001234      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5153 027754 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
5154 027762 005737 001164          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
5155 027766 001414          BEQ      73$           ;BR IF ZERO
5156 027770 013737 001224 001234      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5157 027776 013737 001172 001126      MOV      $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
5158 030004 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
5159 030012 005737 001166          TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
5160 030016 001012          BNE      74$           ;BR IF NOT
5161 030020 012737 177777 001250 73$:    MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
5162 030026 012760 000011 000000      MOV      #11,RPCS1(RO) ;CLEAR THE DRIVE
5163 030034 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
5164 030042 104026          ERROR    26           ;TYPE ERROR MESSAGE 26
5165 030044 013737 001170 001126 74$:    MOV      $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5166 030052 013737 001224 001234      MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
5167 030060 023737 001124 001170      CMP      $GDDAT,$TMP2  ;ALL BITS OK ?
5168 030066 001401          BEQ      75$           ;BR IF OK FROM PORT A.
5169 030070 104007          ERROR    7            ;REPORT ERROR
5170 030072 013737 001172 001126 75$:    MOV      $TMP3,$BDDAT  ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5171 030100 013737 001226 001234      MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
5172 030106 023737 001124 001172      CMP      $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.

```


E08

5173 030114 001401
5174 030116 104007
5175 030120 000240
5176 030122 000004
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228

030124 005737 001274
030124 001406
030130 100002
030132 000137 002602
030134 012737 177777 001274
030140 112737 000023 001102
030146 012737 030176 001106
030154 012737 030176 001110
030162 012737 007640 001176
030170 012706 001100
030202 113760 001224 000010
030210 005060 000012
030214 012760 000011 000000
030222 012760 000013 000000
030230 113760 001226 000010
030236 005060 000012
030242 012760 000011 000000
030250 012760 000013 000000
030256 113760 001226 000010
030264 013737 001226 001236
030272 005760 000000
030276 113760 001224 000010
030304 013737 001224 001234
030312 013737 001224 001240
030320 016037 000012 001126

BEQ 765 ;BR IF OK
ERROR 7 ;REPORT ERROR
765: NOP
15: SCOPE ;LOOP ?

*TEST 23 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
*
*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
* REQUEST'.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1. 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.
*

TST23:
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 #23, \$TSTNM ;TEST NUMBER
MOV #TEST23, \$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST23, \$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000, \$TIMES ;DO 4000. ITERATIONS
TEST23: MOV #STACK, SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA, RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11, RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RPCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB, RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11, RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13, RPCS1(RO) ;RELEASE THE DRIVE

;; *****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB, RPCS2(RO) ;SELECT PORT B
MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RPCS1(RO) ;READ RHCS1
MOVB PORTA, RPCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO), \$BDDAT ;SEE IF DRIVE SEIZED BY PORT B

F08

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 96
 DZRJEA.CMB 02-NOV-76 18:40 T23 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PCRT 'A'

```

5229 030326 010037 001122      MOV      RO,$B0ADR      ;RH11 BASE ADDRESS
5230 030332 062737 000012 001122  ADD      #RPCS1,$B0ADR  ;GENERATE BAD REGISTER ADDRESS
5231 030340 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
5232 030344 023737 001124 001126  CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
5233 030352 001403      BEQ      64$          ;BR IF IT IS
5234 030354 104004      ERROR   4            ;REPORT THE ERROR
5235 030356 000137 031132      JMP      1$          ;BYPASS REST OF THE SUBTEST
5236 030362
5237 030362 113760 001226 000010 64$: MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
5238 030370 013737 001226 001234  MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5239 030376 016037 000012 001126  MOV     RPCS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5240 030404 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5241 030412 013737 001124 001166  MOV     $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5242 030420 005137 001166      COM     $TMP1         ;COMPLEMENT THE EXPECTED STATUS
5243 030424 013737 001126 001164  MOV     $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
5244 030432 043737 001166 001164  BIC     $TMP1,$TMP0   ;CLEAR UNWANTED BITS
5245 030440 023737 001124 001164  CMP     $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5246 030446 001401      BEQ      65$          ;BR IF THEY ARE
5247 030450 104005      ERROR   5            ;REPORT THE ERROR
5248 030452 000240 65$: NOP
5249 030454 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
5250 030462 013737 001224 001234  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5251
5252
5253 ;*****
5254 ;READ RPCS1 THROUGH PORT A - TRY TO SET PORT REQUEST
5255 030470 005037 001244      CLR     CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
5256 030474 016037 000000 001126  MOV     RPCS1(RO),$BDDAT ;GET CONTENTS OF RPCS1
5257 030502 012737 000000 001122  MOV     #RPCS1,$B0ADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5258 030510 060037 001122      ADD     RO,$B0ADR     ;ADD RH11 BASE ADDRESS
5259 030514 005037 001124      CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
5260 030520 013737 001126 001164  MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
5261 030526 042737 173700 001164  BIC     #1C4077,$TMP0 ;SAVE SPECIFIED BITS
5262 030534 023737 001124 001164  CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
5263 030542 001414      BEQ     66$          ;BR IF OK
5264 030544 013737 001126 001174  MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
5265 030552 042737 004077 001174  BIC     #4077,$TMP4   ;CLEAR THE MASKED BITS
5266 030560 053737 001174 001124  BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
5267 030566 104010      ERROR   10          ;REPORT THE ERROR
5268 030570 005137 001244      COM     CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
5269 030574 000240 66$: NOP
5270
5271 ;*****
5272 ;DRIVE SHOULD RETURN TO NEUTRAL
5273
5274
5275 ;RELEASE THE DRIVE FROM PORT B
5276
5277 030576 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
5278 030604 013737 001226 001234  MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5279 030612 012760 000013 000000  MOV     #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT
5280
5281 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5282
5283 030620 005037 001250      CLR     RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
5284 030624 012737 000012 001122  MOV     #RPCS1,$B0ADR ;FORM THE ADDRESS OF RPCS1 FOR TYPEOUT

```

G08

```

5285 030632 060037 001122      ADD      RO,$BDDADR      ;ADD THE I/O BASE ADDRESS
5286 030636 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5287 030644 113760 001224 000010  MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
5288 030652 016037 000012 001170  MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5289 030660 013737 001170 001164  MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5290 030666 042737 100100 001164  BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5291 030674 113760 001226 000010  MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
5292 030702 016037 000012 001172  MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5293 030710 013737 001172 001166  MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5294 030716 042737 100100 001166  BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5295 030724 023737 001164 001166  CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5296 030732 001006      BNE     68$ ;BR IF NOT
5297 030734 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5298 030740 001045      BNE     70$ ;BR IF NOT
5299 030742 104046      ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5300 030744 000137 031130      JMP     72$ ;BYPASS THE REST OF THE CHECKS
5301 030750 013737 001170 001126 68$: MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5302 030756 013737 001226 001234  MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5303 030764 113760 001226 000010  MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
5304 030772 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5305 030776 001414      BEQ     69$ ;BR IF ZERO
5306 031000 013737 001224 001234  MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5307 031006 013737 001172 001126  MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5308 031014 113760 001224 000010  MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
5309 031022 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5310 031026 001012      BNE     70$ ;BR IF NOT
5311 031030 012737 177777 001250 69$: MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5312 031036 012760 000011 000000  MOV     #11,RPCS1(RO) ;CLEAR THE DRIVE
5313 031044 012760 000013 000000  MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
5314 031052 104026      ERROR   26 ;TYPE ERROR MESSAGE 26
5315 031054 013737 001170 001126 70$: MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5316 031062 013737 001224 001234  MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
5317 031070 023737 001124 001170  CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
5318 031076 001401      BEQ     71$ ;BR IF OK FROM PORT A.
5319 031100 104007      ERROR   7 ;REPORT ERROR
5320 031102 013737 001172 001126 71$: MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5321 031110 013737 001226 001234  MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
5322 031116 023737 001124 001172  CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
5323 031124 001401      BEQ     72$ ;BR IF OK
5324 031126 104007      ERROR   7 ;REPORT ERROR
5325 031130 000240 72$: NOP
5326 031132 000004 1$: SCOPE ;LOOP ?

```

5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340

```

*****
*TEST 24 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'
*
*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
* REQUEST'.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1. 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

```

H08

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 98
DZRJEA.CMB 02-NOV-76 18:40 T24 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'

```
5341          : * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5342          : *
5343          : *****
5344 031134      ST24:
5345 031134 005737 0C1274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
5346 031140 001406              BEQ      2$          ;BR IF NOT
5347 031142 100002              SPL      1$          ;BR IF JUST ENTERED TEST
5348 031144 000137 002602      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
5349 031150 012737 177777 001274 1$: MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
5350 031156 112737 000024 001102 2$: MOV     #24,$STNM  ;TEST NUMBER
5351 031164 012737 031206 001106      MOV     #TEST24,$LPADR ;LOAD LOOP ON TEST ADDRESS
5352 031172 012737 031206 001110      MOV     #TEST24,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5353 031200 012737 007640 001176      MOV     #4000,$TIMES  ;DO 4000. ITERATIONS
5354 031206 012706 001100      TEST24: MOV    #STACK,$P ;LOAD THE STACK POINTER
5355
5356          ;CLEAR ATTENTION BITS FOR BOTH PORTS
5357
5358 031212 113760 001224 000010      MOV     PORTA,RPCS2(RO) ;SELECT PORT #A
5359 031220 005060 000012              CLR     RPDS1(RO)      ;SEIZE THE DRIVE
5360 031224 012760 000011 00L J00      MOV     #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
5361 031232 012760 000013 000000      MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
5362 031240 113760 001226 000010      MOV     PORTB,RPCS2(RO) ;SELECT PORT #B
5363 031246 005060 000012              CLR     RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
5364 031252 012760 000011 000000      MOV     #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
5365 031260 012760 000013 000000      MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
5366
5367          ; *****
5368
5369          ;SEIZE THE DRIVE THROUGH PORT A
5370
5371 031266 113760 001224 000010      MOV     PORTA,RPCS2(RO) ;SELECT PORT A
5372 031274 013737 001224 001236      MOV     PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
5373 031302 005760 000000              TST     RPCS1(RO)      ;READ RMCS1
5374 031306 113760 001226 000010      MOV     PORTB,RPCS2(RO) ;SELECT PORT B
5375 031314 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5376 031322 013737 001226 001240      MOV     PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
5377 031330 016037 000012 001126      MOV     RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
5378 031336 010037 001122              MOV     RO,$BDADR     ;R#11 BASE ADDRESS
5379 031342 062737 000012 001122      ADD     #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
5380 031350 005037 001124              CLR     $GDDAT        ;REGISTER SHOULD BE ZERO
5381 031354 023737 001124 001126      CMP     $GDDAT,$BDDAT ;IS THE REGISTER ZERO
5382 031362 001403              BEQ     64$          ;BR IF IT IS
5383 031364 104004              ERROR  4            ;REPORT THE ERROR
5384 031366 000137 032142              JMP     1$          ;BYPASS REST OF THE SUBTEST
5385
5386          64$:
5387 031372 113760 001224 000010      MOV     PORTA,RPCS2(RO) ;SELECT PORT A
5388 031400 013737 001224 001234      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5389 031406 016037 000012 001126      MOV     RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5390 031414 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5391 031422 013737 001124 001166      MOV     $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5392 031430 005137 001166              COM     $TMP1         ;COMPLEMENT THE EXPECTED STATUS
5393 031434 013737 001126 001164      MOV     $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
5394 031442 043737 001166 001164      BIC     $TMP1,$TMP0 ;CLEAR UNWANTED BITS
5395 031450 023737 001124 001164      CMP     $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5396 031456 001401              BEQ     65$          ;BR IF THEY ARE
5397 031460 104005              ERROR  5            ;REPORT THE ERROR
```

```

5397 031462 000240 66$: NOP
5398 031464 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5399 031472 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5400
5401 ;*****
5402 ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
5403
5404 031500 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5405 031504 016037 000000 001126 MOV RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
5406 031512 012737 000000 001122 MOV #RPCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5407 031520 060037 001122 ADD RO,SBADR ;ADD RHI1 BASE ADDRESS
5408 031524 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5409 031530 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5410 031536 042737 173700 001164 BIC #1C4077,$TMP0 ;SAVE SPECIFIED BITS
5411 031544 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5412 031552 001414 BEQ 66$ ;BR IF OK
5413 031554 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
5414 031562 042737 004077 001174 BIC #4077,$TMP4 ;CLEAR THE MASKED BITS
5415 031570 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5416 031576 104010 ERROR IO ;REPORT THE ERROR
5417 031600 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5418 031604 000240 66$: NOP
5419
5420 ;*****
5421 ;DRIVE SHOULD RETURN TO NEUTRAL
5422
5423
5424 ;RELEASE THE DRIVE FROM PORT A
5425
5426 031606 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5427 031614 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5428 031622 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5429
5430 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5431
5432 031630 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5433 031634 012737 000012 001122 MOV #RPDS1,SBADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5434 031642 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
5435 031646 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5436 031654 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5437 031662 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5438 031670 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5439 031676 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5440 031704 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
5441 031712 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5442 031720 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5443 031726 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5444 031734 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5445 031742 001006 BNE 68$ ;BR IF NOT
5446 031744 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5447 031750 001045 BNE 70$ ;BR IF NOT
5448 031752 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5449 031754 000137 032140 JMP 72$ ;BYPASS THE REST OF THE CHECKS
5450 031760 013737 001170 001126 68$: MOV $TMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5451 031766 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5452 031774 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.

```

JOB

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 100
 DZRJEA.CMB 02-NOV-76 18:40 T24 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'

5453	032002	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
5454	032006	001414				BEQ	69\$;BR IF ZERO
5455	032010	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5456	032016	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
5457	032024	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
5458	032032	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
5459	032036	001012				SNE	70\$;BR IF NOT
5460	032040	012737	177777	001250	69\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
5461	032046	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
5462	032054	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
5463	032062	104026				ERROR	26	;TYPE ERROR MESSAGE 26
5464	032064	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5465	032072	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
5466	032100	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
5467	032106	001401				BEQ	71\$;BR IF OK FROM PORT A.
5468	032110	104007				ERROR	7	;REPORT ERROR
5469	032112	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5470	032120	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
5471	032126	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
5472	032134	001401				BEQ	72\$;BR IF OK
5473	032136	104007				ERROR	7	;REPORT ERROR
5474	032140	000240			72\$:	NOP		
5475	032142	000004			1\$:	SCOPE		;LOOP ?

5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496

```

*****
*TEST 25      TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
*
*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 RPDS1.
*  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.
*****
  
```

5497	032144					ST25:		
5498	032144	005737	001274			TST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
5499	032150	001406				BEQ	2\$;BR IF NOT
5500	032152	100002				BPL	1\$;BR IF JUST ENTERED TEST
5501	032154	000137	002602			JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
5502	032160	012737	177777	001274	1\$:	MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR
5503	032166	112737	000025	001102	2\$:	MOVB	#25,\$STNM	;TEST NUMBER
5504	032174	012737	032216	001106		MOV	#TEST25,\$LPADR	;LOAD LOOP ON TEST ADDRESS
5505	032202	012737	032216	001110		MOV	#TEST25,\$LPERR	;LOAD LOOP ON ERROR ADDRESS
5506	032210	012737	007640	001176		MOV	#4000,\$TIMES	;DO 4000. ITERATIONS
5507	032216	012706	001100		TEST25:	MOV	#STACK,\$P	;LOAD THE STACK POINTER
5508								

K08

```

5509                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
5510
5511 032222 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT #A
5512 032230 005060 000012 000000      CLR    RPDS1(RO)      ;SEIZE THE DRIVE
5513 032234 012760 000011 000000      MOV    #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
5514 032242 012760 000013 000000      MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
5515 032250 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT #B
5516 032256 005060 000012 000000      CLR    RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
5517 032262 012760 000011 000000      MOV    #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
5518 032270 012760 000013 000000      MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
5519
5520                                     ;*****
5521
5522                                     ;SEIZE THE DRIVE THROUGH PORT B
5523
5524 032276 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
5525 032304 013737 001226 001236      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
5526 032312 005060 000012 000000      CLR    RPDS1(RO)      ;WRITE RPDS1
5527 032316 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
5528 032324 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5529 032332 013737 001224 001240      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
5530 032340 016037 000012 001126      MOV    RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
5531 032346 010037 001122 001122      MOV    RO,$BDADR ;R#11 BASE ADDRESS
5532 032352 062737 000012 001122      ADD    #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
5533 032360 005037 001124 001126      CLR    $GDDAT ;REGISTER SHOULD BE ZERO
5534 032364 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER ZERO
5535 032372 001403 001124 001126      BEQ    64$ ;BR IF IT IS
5536 032374 104004 001124 001126      ERROR 4 ;REPORT THE ERROR
5537 032376 000137 033350 001126      JMP    1$ ;BYPASS REST OF THE SUBTEST
5538 032402
5539 032402 113760 001226 000010      64$: MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
5540 032410 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5541 032416 016037 000012 001126      MOV    RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5542 032424 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5543 032432 013737 001124 001166      MOV    $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5544 032440 005137 001166 001164      COM    $TMP1 ;COMPLEMENT THE EXPECTED STATUS
5545 032444 013737 001126 001164      MOV    $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
5546 032452 043737 001166 001164      BIC    $TMP1,$TMP0 ;CLEAR UNWANTED BITS
5547 032460 023737 001124 001164      CMP    $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5548 032466 001401 001124 001164      BEQ    65$ ;BR IF THEY ARE
5549 032470 104005 001124 001164      ERROR 5 ;REPORT THE ERROR
5550 032472 000240 001124 001164      65$: NOP
5551
5552                                     ;*****
5553 ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
5554
5555 032474 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
5556 032502 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5557 032510 012760 000013 000000      MOV    #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT A
5558
5559                                     ;*****
5560 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
5561
5562 032516 005037 001244 001126      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5563 032522 016037 000012 001126      MOV    RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
5564 032530 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

L08

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 19:42 PAGE 102
DZRJEA.CMB 02-NOV-76 18:40 T25 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'

```

5565 032536 060037 001122      ADD      RO,$BDADR      ;ADD RH11 BASE ADDRESS
5566 032542 005037 001124      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
5567 032546 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
5568 032554 001403          BEQ      66$           ;BR IF OK
5569 032556 104010          ERROR    10           ;REPORT THE ERROR
5570 032560 005137 001244      COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
5571 032564 016037 000000 001126 66$:      MOV      RPCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
5572 032572 012737 000000 001122      MOV      #RPCS1,$BDADR ;FORM ADDRESS OF REGISTER
5573 032600 060037 001122      ADD      RO,$BDADR    ;ADDRESS BASE
5574 032604 022737 020000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET ?
5575 032612 001404          BEQ      67$           ;BR IF NOT
5576 032614 104011          ERROR    11           ;REPORT THE ERROR
5577 032616 012760 040000 000000      MOV      #TRE,RPCS1(RO) ;CLEAR 'MCPE'
5578 032624 000240          NOP
5579 032626 005737 001244      TST      CKERR        ;WAS RPDS1 NON ZERO ?
5580 032632 001402          BEQ      .+6          ;CONTENTS OF RPDS1 SEEN BY PORT A
5581 032634 000137 033350      JMP      1$           ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

```

;;*****

;RELEASE THE DRIVE FROM PORT B

```

5586          MOV      PORTB,RPCS2(RO) ;SELECT PORT B
5587 032640 113760 001226 000010      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5588 032646 013737 001226 001234      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5589 032654 012760 000013 000000

```

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

```

5593 032662 005037 001250      CLR      RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
5594 032666 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5595 032674 012737 000012 001122      MOV      #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
5596 032702 060037 001122      ADD      RO,$BDADR    ;REGISTER BASE ADDRESS FOR TYPEOUT
5597 032706 113760 001224 000010      MCVB    PORTA,RPCS2(RO) ;SELECT PORT A
5598 032714 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5599 032722 016037 000012 001164      MOV      RPDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT A
5600 032730 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
5601 032736 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5602 032744 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
5603 032752 001404          BEQ      68$           ;BR IF STATUS FROM PORT B ZERO
5604 032754 005737 001164      TST      $TMPD        ;IS STATUS FROM PORT A ZERO ?
5605 032760 001401          BEQ      68$           ;BR IF ZERO
5606 032762 104031          ERROR    31           ;REPORT DRIVE IN NEUTRAL
5607 032764 013737 001164 001126 68$:      MOV      $TMPD,$BDDAT ;CHECK STATUS FROM PORT A
5608 032772 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5609 033000 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
5610 033006 001401          BEQ      69$           ;BR IF OK
5611 033010 104027          ERROR    27           ;REPORT REGISTER ERROR
5612 033012 000240          NOP

```

;RELEASE THE DRIVE FROM PORT A

```

5615          MOV      PORTA,RPCS2(RO) ;SELECT PORT A
5616 033014 113760 001224 000010      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5617 033022 013737 001224 001234      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5618 033030 012760 000013 000000

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

5619
5620

M08

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 103
 DZRJEA.CMB 02-NOV-76 18:40 T25 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'

```

5621
5622 033036 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5623 033042 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5624 033050 060037 001122 ADD RD,$BDADR ;ADD THE I/O BASE ADDRESS
5625 033054 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5626 033062 113760 001224 000010 MOVB PORTA,RPCS2(RD) ;SELECT PORT A.
5627 033070 016037 000012 001170 MOV RPDS1(RD),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5628 033076 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5629 033104 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5630 033112 113760 001226 000010 MOVB PORTB,RPCS2(RD) ;SELECT PORT B.
5631 033120 016037 000012 001172 MOV RPDS1(RD),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5632 033126 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5633 033134 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5634 033142 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5635 033150 001006 BNE 70$ ;BR IF NOT
5636 033152 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5637 033156 001045 BNE 72$ ;BR IF NOT
5638 033160 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5639 033162 000137 033346 JMP 74$ ;BYPASS THE REST OF THE CHECKS
5640 033166 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5641 033174 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5642 033202 113760 001226 001010 MOVB PORTB,RPCS2(RD) ;SELECT PORT B.
5643 033210 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5644 033214 001414 BEQ 71$ ;BR IF ZERO
5645 033216 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5646 033224 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5647 033232 113760 001224 000010 MOVB PORTA,RPCS2(RD) ;SELECT PORT A.
5648 033240 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5649 033244 001012 BNE 72$ ;BR IF NOT
5650 033246 012737 177777 001250 71$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5651 033254 012760 000011 000000 MOV #11,RPCS1(RD) ;CLEAR THE DRIVE
5652 033262 012760 000013 000000 MOV #13,RPCS1(RD) ;RELEASE THE DRIVE
5653 033270 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5654 033272 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5655 033300 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5656 033306 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
5657 033314 001401 BEQ 73$ ;BR IF OK FROM PORT A.
5658 033316 104007 ERROR 7 ;REPORT ERROR
5659 033320 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5660 033326 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5661 033334 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
5662 033342 001401 BEQ 74$ ;BR IF OK
5663 033344 104007 ERROR 7 ;REPORT ERROR
5664 033346 000240 74$: NOP
5665 033350 000004 1$: SCOPE ;LOOP ?

```

```

5666
5667 ;*****
5668 ;*TEST 26 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
5669 ;*
5670 ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
5671 ;* IS SEIZED BY THE OTHER PORT.
5672 ;*
5673 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
5674 ;*
5675 ;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
5676 ;*

```

N08

5677
5678
5679
5680
5681
5682
5683
5684
5685
5686 033352
5687 033352 005737 001274
5688 033356 001406
5689 033360 100002
5690 033362 000137 002602
5691 033366 012737 177777 001274
5692 033374 112737 000026 001102
5693 033402 012737 033424 001106
5694 033410 012737 033424 001110
5695 033416 012737 007640 001176
5696 033424 012706 001100
5697
5698
5699
5700 033430 113760 001224 000010
5701 033436 005060 000012
5702 033442 012760 000011 000000
5703 033450 012760 000013 000000
5704 033456 113760 001226 000010
5705 033464 005060 000012
5706 033470 012760 000011 000000
5707 033476 012760 000013 000000
5708
5709
5710
5711
5712
5713 033504 113760 001224 000010
5714 033512 013737 001224 001236
5715 033520 005060 000012
5716 033524 113760 001226 000010
5717 033532 013737 001226 001234
5718 033540 013737 001226 001240
5719 033546 016037 000012 001126
5720 033554 010037 001122
5721 033560 062737 000012 001122
5722 033566 005037 001124
5723 033572 023737 001124 001126
5724 033600 001403
5725 033602 104004
5726 033604 000137 034556
5727 033610
5728 033610 113760 001224 000010
5729 033616 013737 001224 001234
5730 033624 016037 000012 001126
5731 032632 012737 011700 001124
5732 033640 013737 001124 001166

- * 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.

T26.

```
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
SPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #26,$TSTNM ;TEST NUMBER
MOV #TEST26,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST26,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST26: MOV #STACK,$P ;LOAD THE STACK POINTER
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```
MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
```

;;*****

;SEIZE THE DRIVE THROUGH PORT A

```
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPR ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BDDADR ;RHI1 BASE ADDRESS
ADD #RPDS1,$BDDADR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
BEQ 64$ ;BR IF IT IS
ERROR 4 ;REPORT THE ERROR
JMP 1$ ;BYPASS REST OF THE SUBTEST
```

64\$:

```
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
```

```

033646 005137 001166      COM      $TMP1      ;COMPLEMENT THE EXPECTED STATUS
033652 013737 001126 001164  MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
033660 043737 001166 001164  BIC      $TMP1,$TMP0 ;CLEAR UNWANTED BITS
033666 023737 001124 001164  CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
033674 001401      BEQ      655        ;BR IF THEY ARE
033676 104005      ERROR 5          ;REPORT THE ERROR
033700 000240      655:  NOP

;*****
;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B

033702 113760 001226 000010  MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
033710 013737 001226 001234  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033716 012760 000013 000000  MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PCRT B

;*****
;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A

033724 005037 001244      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
033730 016037 000012 001126  MOV      RPS1(RO), $BDDAT ;GET CONTENTS OF RPS1
033736 012737 000012 001122  MOV      @RPS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033744 060037 001122      ADD     RO,$BADDR   ;ADD RHI1 BASE ADDRESS
033750 005037 001124      CLR     $GDDAT     ;WHAT REGISTER SHOULD BE
033754 023737 001124 001126  CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
033762 001403      BEQ     665        ;BR IF OK
033764 104010      ERROR 10         ;REPORT THE ERROR
033766 005137 001244      COM     CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
033772 016037 000000 001126 655:  MOV      RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RPCS1
034000 012737 000000 001122  MOV      @RPCS1,$BADDR ;FORM ADDRESS OF REGISTER
034006 060037 001122      ADD     RO,$BADDR   ;ADDRESS BASE
034012 032737 020000 001126  BIT      @MCPE,$BDDAT ;IS 'MCPE' SET ?
034020 001404      BEQ     675        ;BR IF NOT
034022 104011      ERROR 11         ;REPORT THE ERROR
034024 012760 040000 000000 675:  MOV      @TRE,RPCS1(RO) ;CLEAR 'MCPE'
034032 000240      NOP
034034 005737 001244      TST     CKERR      ;WAS RPS1 NON ZERO ?
034040 001402      BEQ     +6         ;CONTENTS OF RPS1 SEEN BY PORT B
034042 000137 034556      JMP     IS         ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

;*****
;RELEASE THE DRIVE FROM PORT A

034046 113760 001224 000010  MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
034054 013737 001224 001234  MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034062 012760 000013 000000  MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A

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

034070 005037 001250      CLR      RELERR     ;CLEAR 'RELEASE ERROR' INDICATOR
034074 012737 111700 001124  MOV      @ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
034102 012737 000012 001122  MOV      @RPS1,$BADDR ;REGISTER ADDRESS INCREMENT
034110 060037 001122      ADD     RO,$BADDR   ;REGISTER BASE ADDRESS FOR TYPEOUT
034114 113760 001226 000010  MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
034122 013737 001226 001234  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034130 016037 000012 001164  MOV      RPS1(RO), $TMP0 ;READ STATUS REGISTER FROM PORT B

```

5789	034136	113760	001224	000010		MOV B	PORTA,RPCS2(RO)	:SELECT PORT A
5790	034144	013737	001224	001234		MOV	PORTA,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5791	034152	016037	000012	001126		MOV	RPDS1(RO),%BDDAT	:DRIVE STATUS FROM PORT A
5792	034160	001404				BEQ	68\$:BR IF STATUS FROM PORT A ZERO
5793	034162	005737	001164			TST	\$TMP0	:IS STATUS FROM PORT B ZERO ?
5794	034166	001401				BEQ	68\$:BR IF ZERO
5795	034170	104031				ERRCR	31	:REPORT DRIVE IN NEUTRAL
5796	034172	013737	001164	001126	69\$:	MOV	\$TMP0,%BDDAT	:CHECK STATUS FROM PORT B
5797	034200	013737	001226	001234		MOV	PORTB,PTNBR	:CHANGE PORT ADDRESS FOR TYPEOUT
5799	034206	023737	001124	001126		MOV	%GDDAT,%BDDAT	:COMPARE WITH CONSTANT
5799	034214	001401				BEQ	69\$:BR IF OK
5800	034216	104027				ERROR	27	:REPORT REGISTER ERROR
5801	034220	000240			69\$:	NOP		
5802								
5803								:RELEASE THE DRIVE FROM PORT B
5804								
5805	034222	113760	001226	000010		MOV B	PORTB,RPCS2(RO)	:SELECT PORT B
5806	034230	013737	001226	001234		MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5807	034236	012760	000013	000000		MOV	#13,RPDS1(RO)	:ISSUE RELEASE THROUGH PORT B
5808								
5809								:VERIFY THAT THE DRIVE IS IN NEUTRAL
5810								
5811	034244	005037	001250			CLR	RELERR	:CLEAR THE 'RELEASE ERROR' INDICATOR
5812	034250	012737	000012	001122		MOV	#RPDS1,%BDDAT	:FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5813	034256	060037	001122			ADD	RO,%BDDAT	:ADD THE I/O BASE ADDRESS
5814	034262	012737	011700	001124		MOV	#MOL:PGM:DPR:DRY:VV,%GDDAT	:COMPARISON CONSTANT
5815	034270	113760	001224	000010		MOV B	PORTA,RPCS2(RO)	:SELECT PORT A.
5816	034276	016037	000012	001170		MOV	RPDS1(RO),%TMP2	:GET THE DRIVE STATUS REGISTER FROM PORT A.
5817	034304	013737	001170	001164		MOV	%TMP2,%TMP0	:COPY IT INTO '%TMP0'
5818	034312	042737	100100	001164		BIC	#ATA:VV,%TMP0	:CLEAR PORT DEPENDENT BITS FROM THE COPY
5819	034320	113760	001226	000010		MOV B	PORTB,RPCS2(RO)	:SELECT PORT B.
5820	034326	016037	000012	001172		MOV	RPDS1(RO),%TMP3	:GET THE DRIVE STATUS REGISTER FROM PORT B.
5821	034334	013737	001172	001166		MOV	%TMP3,%TMP1	:COPY IT INTO '%TMP1'
5822	034342	042737	100100	001166		BIC	#ATA:VV,%TMP1	:CLEAR PORT DEPENDENT BITS FROM THE COPY
5823	034350	023737	001164	001166		MOV	%TMP0,%TMP1	:IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5824	034356	001006				BNE	70\$:BR IF NOT
5825	034360	005737	001164			TST	%TMP0	:REGISTERS ARE THE SAME: ARE THEY ZERO ?
5826	034364	001045				BNE	72\$:BR IF NOT
5827	034366	104046				ERROR	46	:REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5828	034370	000137	034554			JMP	74\$:BYPASS THE REST OF THE CHECKS
5829	034374	013737	001170	001126	70\$:	MOV	%TMP2,%BDDAT	:SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5830	034402	013737	001226	001234		MOV	PORTB,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5831	034410	113760	001226	000010		MOV B	PORTB,RPCS2(RO)	:SELECT PORT B.
5832	034416	005737	001164			TST	%TMP0	:SEE IF STATUS EQ 0 FROM PORT A.
5833	034422	001414				BEQ	71\$:BR IF ZERO
5834	034424	013737	001224	001234		MOV	PORTA,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5835	034432	013737	001172	001126		MOV	%TMP3,%BDDAT	: 'BAD DATA' FOR ERROR TYPE OUT
5836	034440	113760	001224	000010		MOV B	PORTA,RPCS2(RO)	:SELECT PORT A.
5837	034446	005737	001166			TST	%TMP1	:SEE IF STATUS EQ ZERO FROM PORT B.
5838	034452	001012				BNE	72\$:BR IF NOT
5839	034454	012737	177777	001250	71\$:	MOV	#-1,RELERR	:SET 'RELEASE ERROR' INDICATOR
5840	034462	012760	000011	000000		MOV	#11,RPDS1(RO)	:CLEAR THE DRIVE
5841	034470	012760	000013	000000		MOV	#13,RPDS1(RO)	:RELEASE THE DRIVE
5842	034476	104026				ERROR	26	:TYPE ERROR MESSAGE 26
5843	034500	013737	001170	001126	72\$:	MOV	%TMP2,%BDDAT	:LOOK FOR BIT FAILURES WHEN RPDS1 READ
5844	034506	013737	001224	001234		MOV	PORTA,PTNBR	:CHANGE PORT NUMBER

```

5845 034514 023737 001124 001170      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
5846 034522 001401                      BEQ      735                ;BR IF OK FROM PORT A.
5847 034524 104007                      ERROR    7                  ;REPORT ERROR
5848 034526 013737 001172 001126 735:  MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5849 034534 013737 001226 001234      MOV      PORTB,PTNBR       ;CHANGE PORT NUMBER
5850 034542 023737 001124 001172      CMP      $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
5851 034550 001401                      BEQ      745                ;BR IF OK
5852 034552 104007                      ERROR    7                  ;REPORT ERROR
5853 034554 000240 745:  NOP
5854 034556 000004 15:  SCOPE                        ;LOOP ?

```

```

*****
*TEST 27      TEST SEIZE BY WRITING ATTENTION BIT
*
*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
* (RPAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER
* PORT.
*
* A. WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; 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.
*****

```

```

5871
5872 034560
5873 034560 005737 001274      TST      KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
5874 034564 001406                      BEQ      15                ;BR IF NOT
5875 034566 100002                      BPL      15                ;BR IF JUST ENTERED TEST
5876 034570 000137 002602      JMP      EXEC              ;RETURN & GET NEXT TEST NUMBER
5877 034574 012737 177777 001274 15:  MOV      #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
5878 034602 112737 000027 001102 25:  MOVB     #27,$STNM         ;TEST NUMBER
5879 034610 012737 034632 001106      MOV      #TEST27,$LPADR    ;LOAD LOOP ON TEST ADDRESS
5880 034616 012737 034632 001110      MOV      #TEST27,$LPERA    ;LOAD LOOP ON ERROR ADDRESS
5881 034624 012737 007640 001176      MOV      #4000,$TIMES      ;DO 4000. ITERATIONS
5882 034632 012706 001100      TEST27: MOV     #STACK,SP    ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

5883
5884
5885
5886 034636 113760 001224 000010      MOVB     PORTA,RPCS2(RO)   ;SELECT PORT #A
5887 034644 005060 000012                      CLR      RPDS1(RO)        ;SEIZE THE DRIVE
5888 034650 012760 000011 000000      MOV      #11,RPCS1(RO)    ;ISSUE DRIVE CLEAR
5889 034656 012760 000013 000000      MOV      #13,RPCS1(RO)    ;RELEASE THE DRIVE
5890 034664 113760 001226 000010      MOVB     PORTB,RPCS2(RO)   ;SELECT PORT #B
5891 034672 005060 000012                      CLR      RPDS1(RO)        ;SEIZE THE DRIVE THROUGH PORT 'B'
5892 034676 012760 000011 000000      MOV      #11,RPCS1(RO)    ;ISSUE DRIVE CLEAR
5893 034704 012760 000013 000000      MOV      #13,RPCS1(RO)    ;RELEASE THE DRIVE

```

```

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

```

```

5894
5895
5896
5897
5898 034712 113760 001230 000010      MOVB     PORTC,RPCS2(RO)   ;SELECT DRIVE NOT BEING TESTED
5899 034720 013737 001224 001236      MOV      PORTA,SEIZPT     ;'SEIZED' PORT ADDRESS
5900

```

E09

```

5901                                     ;*****
5902                                     ;WRITE THE DRIVE'S ATTENTION BIT
5903
5904 034726 013760 001232 000016          MOV   ASR1,RPAS(RO) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
5905 034734 113760 001224 000010          MOVB  PORTA,RPCS2(RO) ;SELECT PORT A
5906 034742 013737 001224 001234          MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5907
5908                                     ;*****
5909                                     ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
5910
5911 034750 005760 000012                    TST   RPDS1(RO) ;SEE THE REGISTER THROUGH PORT A ?
5912 034754 001014                          BNE   1$ ;BR IF YES
5913 034756 113760 001226 000010          MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
5914 034764 013737 001226 001234          MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5915 034772 005760 000012                    TST   RPDS1(RO) ;SEE REGISTER THROUGH PORT B ?
5916 034776 001021                          BNE   2$ ;BR IF YES
5917 035000 104042                          ERROR 42 ;DRIVE NOT IN NEUTRAL OR SEIZED
5918 035002 000137 036552                    JMP   4$ ;BYPASS REST OF TEST
5919 035006
5920 035006 113760 001226 000010          1$: MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
5921 035014 013737 001226 001234          MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5922 035022 005760 000012                    TST   RPDS1(RO) ;REGISTER SHOULD BE ZERO THROUGH PORT B
5923 035026 001002                          BNE   +6 ;BR IF STATUS REG IS NOT ZERO
5924 035030 000137 035702                    JMP   3$ ;STATUS REG IS ZERO
5925 035034 104043                          ERROR 43 ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
5926 035036 000137 036552                    JMP   4$ ;BYPASS REST OF TEST
5927
5928                                     ;*****
5929                                     ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
5930
5931                                     2$:
5932 035042 005037 001244                      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5933 035046 016037 000012 001126          MOV   RPDS1(RO),SBDAT ;GET CONTENTS OF RPDS1
5934 035054 012737 000012 001122          MOV   #RPDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5935 035062 060037 001122                    ADD   RO,SBDADR ;ADD RHI1 BASE ADDRESS
5936 035066 012737 011700 001124          MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
5937 035074 013737 001126 001164          MOV   SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
5938 035102 042737 106077 001164          BIC   #1C71700,STMP0 ;SAVE SPECIFIED BITS
5939 035110 023737 001124 001164          CMP   $GDDAT,STMP0 ;COMPARE THE BITS
5940 035116 001414                          BEQ   64$ ;BR IF OK
5941 035120 013737 001126 001174          MOV   SBDAT,STMP4 ;COPY 'BAD DATA'
5942 035126 042737 071700 001174          BIC   #71700,STMP4 ;CLEAR THE MASKED BITS
5943 035134 052737 001174 001124          BIS   STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5944 035142 104010                          ERROR 10 ;REPORT THE ERROR
5945 035144 005137 001244                      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5946 035150 000240
5947 035152 013737 001226 001236          64$: NOP
5948 035160 013737 001224 001240          MOV   PORTB,SEIZPT ;ADDRESS FOR ERROR MESSAGE
5949
5950                                     ;RELEASE THE DRIVE FROM PORT B
5951
5952 035166 113760 001226 000010          MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
5953 035174 013737 001226 001234          MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5954 035202 012760 000013 000000          MOV   #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5955
5956                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B

```


G09

6013	035566	005737	001166		TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
6014	035572	001012			BNE	70\$;BR IF NOT
6015	035574	012737	177777	001250	69\$:	MOV	#-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6016	035602	012760	000011	000000		MOV	#11,RPCS1(RO) ;CLEAR THE DRIVE
6017	035610	012760	000013	000000		MOV	#13,RPCS1(RO) ;RELEASE THE DRIVE
6018	035616	104026				ERROR	26 ;TYPE ERROR MESSAGE 26
6019	035620	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT ;LOOK FOR BIT FAILURES WHEN RFD51 READ
6020	035626	013737	001224	001234		MOV	PORTA,PTNBR ;CHANGE PORT NUMBER
6021	035634	023737	001124	001170		CMP	\$GDDAT,\$TMP2 ;ALL BITS OK ?
6022	035642	001401				BEG	71\$;BR IF OK FROM PORT A.
6023	035644	104007				ERROR	7 ;REPORT ERROR
6024	035646	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT ;CHECK RFD51 FOR BIT FAILURES - FROM PORT B.
6025	035654	013737	001226	001234		MOV	PORTB,PTNBR ;CHANGE PORT NUMBER
6026	035662	023737	001124	001172		CMP	\$GDDAT,\$TMP3 ;SEE IF READ OK FROM PORT B.
6027	035670	001401				BEG	72\$;BR IF OK
6028	035672	104007				ERROR	7 ;REPORT ERROR
6029	035674	000240			72\$:	NOP	
6030	035676	000137	036552			JMP	4\$
6031							
6032							
6033							*****
6034							;THE DRIVE IS SEIZED BY PORT A. VERIFY THAT PORT B HAS PORT REQUEST SET
6035	035702				3\$:		
6036	035702	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
6037	035710	013737	001224	001234		MOV	PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6038	035716	005037	001244			CLR	CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6039	035722	016037	000012	001126		MOV	RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
6040	035730	012737	000012	001122		MOV	#RPDS1,\$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6041	035736	060037	001122			ADD	RO,\$BDAOR ;ADD RH11 BASE ADDRESS
6042	035742	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT ;WHAT REGISTER SHOULD BE
6043	035750	013737	001126	001164		MOV	\$BDDAT,\$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
6044	035756	042737	106077	001164		BIC	#1C71700,\$TMP0 ;SAVE SPECIFIED BITS
6045	035764	023737	001124	001164		CMP	\$GDDAT,\$TMP0 ;COMPARE THE BITS
6046	035772	001414				BEG	73\$;BR IF OK
6047	035774	013737	001126	001174		MOV	\$BDDAT,\$TMP4 ;COPY 'BAD DATA'
6048	036002	042737	071700	001174		BIC	#71700,\$TMP4 ;CLEAR THE MASKED BITS
6049	036010	053737	001174	001124		BIS	\$TMP4,\$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6050	036016	104010				ERROR	10 ;REPORT THE ERROR
6051	036020	005137	001244			COM	CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6052	036024	000240			73\$:	NOP	
6053	036026	013737	001224	001236		MOV	PORTA,SEIZPT ;ADDRESS FOR ERROR MESSAGE
6054	036034	013737	001226	001240		MOV	PORTB,OPPRT ;SAME AS ABOVE
6055							
6056							;RELEASE THE DRIVE FROM PORT A
6057							
6058	036042	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
6059	036050	013737	001224	001234		MOV	PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6060	036056	012760	000013	000000		MOV	#13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
6061							
6062							;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
6063							
6064	036064	005037	001250			CLR	RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
6065	036070	012737	111700	001124		MOV	#ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
6066	036076	012737	000012	001122		MOV	#RPDS1,\$BDAOR ;REGISTER ADDRESS INCREMENT
6067	036104	060037	001122			ADD	RO,\$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
6068	036110	113760	001226	000010		MOVB	PORTB,RPCS2(RO) ;SELECT PORT B


```

6125 036474 013737 001170 001126 79$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6126 036502 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6127 036510 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6128 036516 001401 BEQ B0$ ;BR IF OK FROM PORT A.
6129 036520 104007 ERROR 7 ;REPORT ERROR
6130 036522 013737 001172 001126 80$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6131 036530 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6132 036536 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6133 036544 001401 BEQ B1$ ;BR IF OK
6134 036546 104007 ERROR 7 ;REPORT ERROR
6135 036550 000240 81$: NOP
6136 036552 000004 4$: SCOPE ;LOOP ?

```

```

6137
6138
6139
6140 ;*****
6141 ;*TEST 30 TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
6142 ;*
6143 ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
6144 ;* THE DRIVE'S ATTENTION BIT.
6145 ;*
6146 ;* A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
6147 ;* BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
6148 ;*
6149 ;* B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
6150 ;*
6151 ;*****

```

```

6151 036554
6152 036554 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6153 036560 001406 BEQ 2$ ;BR IF NOT
6154 036562 100002 BPL 1$ ;BR IF JUST ENTERED TEST
6155 036564 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6156 036570 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6157 036576 112737 000030 001102 2$: MOVB #30,$STNM ;TEST NUMBER
6158 036604 012737 036626 001106 MOV #TEST30,$LPADR ;LOAD LOOP ON TEST ADDRESS
6159 036612 012737 036626 001110 MOV #TEST30,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6160 036620 012737 007640 001176 MOV #4000.,$TIMES ;DO 4000. ITERATIONS
6161 036626 012706 001100 TEST30: MOV #STACK,$P ;LOAD THE STACK POINTER

```

```

6162
6163 ;CLEAR ATTENTION BITS FOR BOTH PORTS
6164
6165 036632 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
6166 036640 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
6167 036644 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6168 036652 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6169 036660 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
6170 036666 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
6171 036672 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6172 036700 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6173 036706 113760 001230 000010 MOVB PORTC,RPCS2(RO) ;SELECT DRIVE NOT BEING TESTED
6174
6175
6176 ;*****
6177 ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST

```

```

6178 036714 013737 001232 001164 MOV ASR1,$TMP0 ;STORE ATTN BIT FOR PORT A
6179 036722 005137 001164 COM $TMP0 ;COMPLEMENT IT
6180 036726 013760 001164 000016 MOV $TMP0,RPAS(RO) ;WRITE THE ATTN REGISTER

```

6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236

:VERIFY THAT DRIVE REMAINED IN NEUTRAL

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

6188 036734 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
6189 036740 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPECUT
6190 036746 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
6191 036752 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6192 036760 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
6193 036766 016037 000012 001170 MOV RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6194 036774 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6195 037002 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6196 037010 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
6197 037016 016037 000012 001172 MOV RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6198 037024 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
6199 037032 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6200 037040 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6201 037046 001006 BNE 64$ ;BR IF NOT
6202 037050 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6203 037054 001045 BNE 66$ ;BR IF NOT
6204 037056 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6205 037060 000137 037244 JMP 68$ ;BYPASS THE REST OF THE CHECKS
6206 037064 013737 001170 001126 64$: MOV $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6207 037072 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6208 037100 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
6209 037106 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6210 037112 001414 BEQ 65$ ;BR IF ZERO
6211 037114 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6212 037122 013737 001172 001126 MOV $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
6213 037130 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
6214 037136 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6215 037142 001012 BNE 66$ ;BR IF NOT
6216 037144 012737 177777 001250 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6217 037152 012760 000011 000000 MOV #11,RPCS1(R0) ;CLEAR THE DRIVE
6218 037160 012760 000013 000000 MOV #13,RPCS1(R0) ;RELEASE THE DRIVE
6219 037166 104021 ERROR 21 ;TYPE ERROR MESSAGE 21
6220 037170 013737 001170 001126 66$: MOV $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6221 037176 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6222 037204 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6223 037212 001401 BEQ 67$ ;BR IF OK FROM PORT A.
6224 037214 104007 ERROR 7 ;REPORT ERROR
6225 037216 013737 001172 001126 67$: MOV $TMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6226 037224 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6227 037232 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6228 037240 001401 BEQ 68$ ;BR IF OK
6229 037242 104007 ERROR 7 ;REPORT ERROR
6230 037244 000240 68$: NOP
6231 037246 000004 SCOPE ;LOOP ?

```

*TEST 31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
*

K09

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 114
DZRJEA.CMB 02-NOV-76 18:40 T31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE

```
6237 ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
6238 ;*
6239 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
6240 ;*
6241 ;* B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
6242 ;*
6243 ;* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
6244 ;* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR
6245 ;* PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
6246 ;*
6247 ;*****
6248 TST31: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6249 037250 005737 001274 BEQ 2$ ;BR IF NOT
6250 037254 001406 BPL 1$ ;BR IF JUST ENTERED TEST
6251 037256 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6252 037260 000137 002602 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6253 037264 012737 177777 001274 2$: MOVB #31,$TSTNM ;TEST NUMBER
6254 037272 112737 000031 001102 MOV #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
6255 037300 012737 037322 001106 MOV #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6256 037306 012737 037322 001110 MOV #4,$TIMES ;DO 4 ITERATIONS
6257 037314 012737 000004 001175 TEST31: MOV #STACK,$P ;LOAD THE STACK POINTER
6258 037322 012706 001100 ;CLEAR ATTENTION BITS FOR BOTH PORTS
6259
6260
6261 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
6262 037326 113760 001224 000010 CLR RPDS1(RO) ;SEIZE THE DRIVE
6263 037334 005060 000012 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6264 037340 012760 000011 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6265 037346 012760 000013 000000 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
6266 037354 113760 001226 000010 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
6267 037362 005060 000012 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6268 037366 012760 000011 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6269 037374 012760 000013 000000 ;*****
6270
6271 ;SEIZE THE DRIVE THROUGH PORT A
6272
6273 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
6274 037402 113760 001224 000010 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
6275 037410 013737 001224 001236 CLR RPDS1(RO) ;WRITE RPDS1
6276 037416 005060 000012 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
6277 037422 013737 001226 001240 ;*****
6278
6279 ;FORCE AN ERROR
6280
6281 MOV #-1,RPER1(RO) ;SET ERROR BITS
6282 037430 012760 177777 000014 ;*****
6283
6284 ;START THE TIMER
6285
6286 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
6287 037436 005037 001252 MOV #2000,WATCH ;SET WATCH TO 2000 MS
6288 037442 012737 003720 001254 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
6289 037450 113760 001226 000010 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
6290 037456 013737 001226 001234 ;*****
6291
6292
```

```

6293                                     ;WAIT FOR DRIVE TO TIMEOUT
6294
6295 037464 005760 000012 1$: TST RPDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
6296 037470 001004 BNE 2$ ;BR IF DRIVE RELEASED
6297 037472 005737 001254 TST WATCH ;WATCH AT ZERO ?
6298 037476 001372 BNE 1$ ;BR IF NOT
6299 037500 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
6300 037502
6301 037502 113760 001224 000010 2$: MOVB PORTA,RPCS2(RO) ;SELECT PORT A
6302 037510 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6303
6304 ;*****
6305 ;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET
6306
6307 037516 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6308 037522 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
6309 037530 012737 000012 001122 MOV #RPDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6310 037536 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
6311 037542 012737 040000 001124 MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
6312 037550 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
6313 037556 042737 137777 001164 BIC #1C4000, $TMP0 ;SAVE SPECIFIED BITS
6314 037564 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6315 037572 001414 BEQ 66$ ;BR IF OK
6316 037574 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6317 037602 042737 040000 001174 BIC #40000, $TMP4 ;CLEAR THE MASKED BITS
6318 037610 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6319 037616 104023 ERROR 23 ;TYPE MESSAGE 23
6320 037620 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6321 037624 000240 66$: NOP
6322
6323 ;*****
6324 ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6325
6326 037626 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6327 037632 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
6328 037640 012737 000014 001122 MOV #RPER1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6329 037646 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
6330 037652 012737 177777 001124 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
6331 037660 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
6332 037666 001403 BEQ 68$ ;BR IF OK
6333 037670 104010 ERROR 10 ;REPORT THE ERROR
6334 037672 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6335 037676 000240 68$: NOP
6336
6337 ;*****
6338 ;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET
6339
6340 037700 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6341 037704 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
6342 037712 012737 000012 001122 MOV #RPDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6343 037720 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
6344 037724 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6345 037732 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
6346 037740 042737 077777 001164 BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
6347 037746 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6348 037754 001414 BEQ 70$ ;BR IF OK

```

M09

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 16:42 PAGE 116
 DZRJEA.CMB 02-NOV-76 18:40 T31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE

6349	037756	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
6350	037764	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
6351	037772	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
6352	040000	104041				ERROR	41	;TYPE MESSAGE 41
6353	040002	005137	0C1244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6354	040006	000240			70\$:	NOP		
6355								
6356								
6357								
6358								
6359								
E360								
6361	040010	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
6362	040014	012737	000012	001122		MOV	#RPDS1,\$BDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6363	040022	060037	001122			ADD	RO,\$BDADR	;ADD THE I/O BASE ADDRESS
6364	040026	012737	051700	001124		MOV	#51700,\$GDDAT	;COMPARISON CONSTANT
6365	040034	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
6366	040042	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
6367	040050	013737	001170	001164		MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
6368	040056	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6369	040064	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
6370	040072	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
6371	040100	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
6372	040106	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6373	040114	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6374	040122	001006				BNE	72\$;BR IF NOT
6375	040124	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6376	040130	001045				BNE	74\$;BR IF NOT
6377	040132	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6378	040134	000137	040334			JMP	76\$;BYPASS THE REST OF THE CHECKS
6379	040140	013737	001170	001126	72\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6380	040146	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL.
6381	040154	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
6382	040162	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
6383	040166	001414				BEQ	73\$;BR IF ZERO
6384	040170	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6385	040176	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
6386	040204	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
6387	040212	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
6388	040216	001012				BNE	74\$;BR IF NOT
6389	040220	012737	177777	001250	73\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
6390	040226	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
6391	040234	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
6392	040242	104026				ERROR	26	;TYPE ERROR MESSAGE 26
6393	040244	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6394	040252	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
6395	040260	042737	100000	001170		BIC	#ATA,\$TMP2	;DON'T CHECK THE ATTN BIT
6396	040266	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
6397	040274	001401				BEQ	75\$;BR IF OK FROM PORT A.
6398	040276	104007				ERROR	7	;REPORT ERROR
6399	040300	013737	001172	001126	75\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6400	040306	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
6401	040314	042737	100000	001172		BIC	#ATA,\$TMP3	;DON'T CHECK THE ATTN BIT
6402	040322	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
6403	040330	001401				BEQ	76\$;BR IF OK
6404	040332	104007				ERROR	7	;REPORT ERROR

;*****

;VERIFY THAT THE DRIVE IS IN NEUTRAL

6405 040334 000240
6406
6407
6408
6409
6410 040336 113760 001226 000010
6411 040344 013737 001226 001234
6412 040352 005037 001244
6413 040356 016037 000012 001126
6414 040364 012737 000012 001122
6415 040372 060037 001122
6416 040376 005037 001124
6417 040402 013737 001126 001164
6418 040410 042737 077777 001164
6419 040416 023737 001124 001164
6420 040424 001414
6421 040426 013737 001126 001174
6422 040434 042737 100000 001174
6423 040442 053737 001174 001124
6424 040450 104052
6425 040452 005137 001244
6426 040455 000240
6427
6428
6429
6430 040460 113760 001224 000010
6431 040466 005060 000012
6432 040472 012760 000011 000000
6433 040500 012760 000013 000000
6434 040506 000004
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450 040510
6451 040510 005737 001274
6452 040514 001406
6453 040516 100002
6454 040520 000137 002602
6455 040524 012737 177777 001274
6456 040532 112737 000032 001102
6457 040540 012737 040562 001106
6458 040546 012737 040562 001110
6459 040554 012737 000004 001176
6460 040562 012706 001100

76\$: NOP
;*****
;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET
MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),\$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,\$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,\$B0ADR ;ADD RH11 BASE ADDRESS
CLR \$GDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT,\$TMPD ;MOVE REGISTER CONTENTS TO '\$TMPD'
BIC #1CATA,\$TMPD ;SAVE SPECIFIED BITS
CMP \$GDDAT,\$TMPD ;COMPARE THE BITS
BEQ 77\$;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 52 ;TYPE MESSAGE 52
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
77\$: NOP
;CLEAR ATTENTION BIT FOR PORT A
MOV B PORTA,RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
3\$: SCOPE ;LOOP ?
;*****
*TEST 32 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 RPDS1.
*
* B. WRITE 1'S INTO RPER1 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.
*
;*****
†ST32:
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\$: MOV #32,\$STNM ;TEST NUMBER
MOV #TEST32,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST32,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,\$TIMES ;DO 4 ITERATIONS
TEST32: MOV #STACK,SP ;LOAD THE STACK POINTER

C10

```

6517 041032 001414 BEQ 66$ ;BR IF OK
6518 041034 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6519 041042 042737 040000 001174 BIC #40000,$TMP4 ;CLEAR THE MASKED BITS
6520 041050 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6521 041056 104023 ERROR 23 ;TYPE MESSAGE 23
6522 041060 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6523 041064 000240 66$: NOP
6524
6525 ;*****
6526 ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6527
6528 041066 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6529 041072 016037 000014 001126 MOV RPER1(RO),$BDDAT ;GET CONTENTS OF RPER1
6530 041100 012737 000014 001122 MOV #RPER1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6531 041106 060037 001122 ADD RO,$BADDR ;ADD RHI1 BASE ADDRESS
6532 041112 012737 177777 001124 MOV #177777,$GDDAT ;WHAT REGISTER SHOULD BE
6533 041120 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
6534 041126 001403 BEQ 68$ ;BR IF OK
6535 041130 104010 ERROR 10 ;REPORT THE ERROR
6536 041132 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6537 041136 000240 68$: NOP
6538
6539 ;*****
6540 ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
6541
6542 041140 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6543 041144 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
6544 041152 012737 000012 001122 MOV #RPDS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6545 041160 060037 001122 ADD RO,$BADDR ;ADD RHI1 BASE ADDRESS
6546 041164 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6547 041172 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
6548 041200 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
6549 041206 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6550 041214 001414 BEQ 70$ ;BR IF OK
6551 041216 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6552 041224 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6553 041232 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6554 041240 104041 ERROR 41 ;TYPE MESSAGE 41
6555 041242 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6556 041246 000240 70$: NOP
6557
6558
6559 ;*****
6560
6561 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6562
6563 041250 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
6564 041254 012737 000012 001122 MOV #RPDS1,$BADDR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6565 041262 060037 001122 ADD RO,$BADDR ;ADD THE I/O BASE ADDRESS
6566 041266 012737 051700 001124 MOV #51700,$GDDAT ;COMPARISON CONSTANT
6567 041274 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
6568 041302 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6569 041310 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
6570 041316 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6571 041324 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
6572 041332 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

D10

```

6573 041340 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO 'TMP1'
6574 041346 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6575 041354 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6576 041362 001006                BNE      72$             ;BR IF NOT
6577 041364 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6578 041370 001045                BNE      74$             ;BR IF NOT
6579 041372 104046                ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6580 041374 000137 041574                JMP      76$             ;BYPASS THE REST OF THE CHECKS
6581 041400 013737 001170 001126 72$:      MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6582 041406 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6583 041414 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
6584 041422 005737 001164                TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6585 041426 001414                BEQ      73$             ;BR IF ZERO
6586 041430 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6587 041436 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6588 041444 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
6589 041452 005737 001166                TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6590 041456 001012                BNE      74$             ;BR IF NOT
6591 041460 012737 177777 001250 73$:      MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
6592 041466 012760 000011 000000      MOV      #11,RPCS1(RO)  ;CLEAR THE DRIVE
6593 041474 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
6594 041502 104026                ERROR    26             ;TYPE ERROR MESSAGE 26
6595 041504 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6596 041512 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
6597 041520 042737 100000 001170      BIC      #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
6598 041526 023737 001124 001170      CMP      $GDDAT,$TMP2   ;ALL BITS OK ?
6599 041534 001401                BEQ      75$             ;BR IF OK FROM PORT A.
6600 041536 104007                ERROR    7              ;REPORT ERROR
6601 041540 013737 001172 001126 75$:      MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6602 041546 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
6603 041554 042737 100000 001172      BIC      #ATA,$TMP3     ;DON'T CHECK THE ATTN BIT
6604 041562 023737 001124 001172      CMP      $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
6605 041570 001401                BEQ      76$             ;BR IF OK
6606 041572 104007                ERROR    7              ;REPORT ERROR
6607 041574 000240                NOP
6608
6609
6610
6611
6612 041576 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
6613 041604 013737 001224 001234      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6614 041612 005037 001244                CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6615 041616 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
6616 041624 012737 000012 001122      MOV      #RPDS1,$BADDR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6617 041632 060037 001122                ADD      RO,$BADDR       ;ADD RHI1 BASE ADDRESS
6618 041636 005037 001124                CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
6619 041642 013737 001126 001164      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO 'TMP0'
6620 041650 042737 077777 001164      BIC      #!CATA,$TMP0    ;SAVE SPECIFIED BITS
6621 041656 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
6622 041664 001414                BEQ      77$             ;BR IF OK
6623 041666 013737 001126 001174      MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
6624 041674 042737 100000 001174      BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
6625 041702 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6626 041710 104052                ERROR    52             ;TYPE MESSAGE 52
6627 041712 005137 001244                COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6628 041716 000240                NOP
77$:

```

E10

6629
6630
6631
6632 041720 113760 001226 000010
6633 041726 005060 000012
6634 041732 012760 000011 000000
6635 041740 012760 000013 000000
6636 041746 000004
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656
6657
6658
6659 041750
6660 041750 005737 001274
6661 041754 001406
6662 041756 100002
6663 041760 000137 002602
6664 041764 012737 177777 001274
6665 041772 112737 000033 001102
6666 042000 012737 042022 001106
6667 042006 012737 042022 001110
6668 042014 012737 007640 001176
6669 042022 012706 001100
6670
6671
6672
6673 042026 113760 001224 000010
6674 042034 005060 000012
6675 042040 012760 000011 000000
6676 042046 012760 000013 000000
6677 042054 113760 001226 000010
6678 042062 005060 000012
6679 042066 012760 000011 000000
6680 042074 012760 000013 000000
6681
6682
6683
6684

```

;CLEAR ATTENTION BIT FOR PORT B
MOV B PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
;LOOP ?
35: SCOPE

;*****
*TEST 33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
*
*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 RPDS1.
*
* B. WRITE 1'S INTO RPER1 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 RPER1 HAS NOT BEEN CLEARED.
*
* D. CLEAR RPER1 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.
*
;*****
†ST33:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ ZS ;BR IF NOT
BPL IS ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #33,$STNM ;TEST NUMBER
MOV #TEST33,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST33,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST33: MOV #STACK,SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOV B PORTA,RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
MOV B PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
;*****
;SEIZE THE DRIVE THROUGH PORT A

```


G10

MD-11-DRIE-A. DUAL CONTR. LOGIC TEST - PART 1 MACY!! 27(1006) 02-NOV-76 18:42 PAGE 123
 DRIE.A.CMB 02-NOV-76 19:40 T33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

6741	042446	012737	177777	001124		MOV	#177777,\$GDDAT	;WHAT REGISTER SHOULD BE
6742	042454	023737	001124	001126		CMF	\$GDDAT,\$BDDAT	;IS THE REGISTER OK ?
6743	042462	001403				BEQ	70\$;BR IF OK
6744	042464	104010				ERROR	10	;REPORT THE ERROR
6745	042466	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6746	042472	000240			70\$:	NOP		
6747								
6748								
6749								
6750								
6751	042474	012760	000011	000000		MOV	#11,RPCS1(RO)	;ISSUE A DRIVE CLEAR
6752								
6753								
6754								
6755								
6756								
6757	042502	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A
6758	042510	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6759	042516	012760	000013	000000		MOV	#13,RPCS1(RO)	;ISSUE RELEASE THROUGH PORT A
6760								
6761								
6762								
6763	042524	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
6764	042530	012737	000012	001122		MOV	#RPDS1,\$BDDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6765	042536	060037	001122			ADD	RO,\$BDDADR	;ADD THE I/O BASE ADDRESS
6766	042542	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
6767	042550	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
6768	042556	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
6769	042564	013737	001170	001164		MOV	\$TMP2,\$TMP0	;COPY IT INTO 'TMP0'
6770	042572	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6771	042600	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
6772	042606	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
6773	042614	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO 'TMP1'
6774	042622	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6775	042630	023737	001164	001166		CMF	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6776	042636	001006				BNE	72\$;BR IF NOT
6777	042640	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6778	042644	001045				BNE	74\$;BR IF NOT
6779	042646	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6780	042650	000137	043034			JMP	76\$;BYPASS THE REST OF THE CHECKS
6781	042654	013737	001170	001126	72\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6782	042662	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6783	042670	113760	001226	000010		MOV	PORTB,RPCS2(RO)	;SELECT PORT B.
6784	042676	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
6785	042702	001414				BEQ	73\$;BR IF ZERO
6786	042704	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6787	042712	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
6788	042720	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A.
6789	042726	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
6790	042732	001012				BNE	74\$;BR IF NOT
6791	042734	012737	177777	001250	73\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
6792	042742	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
6793	042750	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
6794	042756	104026				ERROR	26	;TYPE ERROR MESSAGE 26
6795	042760	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6796	042766	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER

H10

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 124
DZRJEA.CMB 02-NOV-76 18:40 T33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

6797	042774	023737	001124	001170	CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
6798	043002	001401			BEQ	75\$;BR IF OK FROM PORT A.
6799	043004	104007			ERROR	7	;REPORT ERROR
6800	043006	013737	001172	001126	75\$: MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6801	043014	013737	001226	001234	MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
6802	043022	023737	001124	001172	CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT E.
6803	043030	001401			BEQ	76\$;BR IF OK
6804	043032	104007			ERROR	7	;REPORT ERROR
6805	043034	000240			76\$: NOP		
6806	043036	000004			2\$: SCOPE		;LOOP ?

6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852

```
*****  
*TEST 34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET  
*  
*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 RPDS1.  
*  
* B. WRITE 1'S INTO RPER1 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 RPER1 HAS NOT BEEN CLEARED.  
*  
* D. CLEAR RPER1 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.  
*  
*****
```

6828	043040				†ST34:	TST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
6829	043040	005737	001274			BEG	2\$;BR IF NOT
6830	043044	001406				BPL	1\$;BR IF JUST ENTERED TEST
6831	043046	100002				JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
6832	043050	000137	002602		1\$: MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR	
6833	043054	012737	177777	001274	2\$: MOV	#34,\$STNM	;TEST NUMBER	
6834	043062	112737	000034	001102		MOV	#TEST34,\$LPADR	;LOAD LOOP ON TEST ADDRESS
6835	043070	012737	043112	001106		MOV	#TEST34,\$LPERR	;LOAD LOOP ON ERROR ADDRESS
6836	043076	012737	043112	001110		MOV	#4000,\$TIMES	;DO 4000. ITERATIONS
6837	043104	012737	007640	001176	TEST34:	MOV	#STACK,SP	;LOAD THE STACK POINTER
6838	043112	012706	001100					
6839								
6840								
6841								
6842	043116	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT #A
6843	043124	005060	000012			CLR	RPDS1(RO)	;SEIZE THE DRIVE
6844	043130	012760	000011	000000		MOV	#11,RPCS1(RO)	;ISSUE DRIVE CLEAR
6845	043136	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
6846	043144	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT #B
6847	043152	005060	000012			CLR	RPDS1(RO)	;SEIZE THE DRIVE THROUGH PORT 'B'
6848	043156	012760	000011	000000		MOV	#11,RPCS1(RO)	;ISSUE DRIVE CLEAR
6849	043164	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE

```
;;*****  
;SEIZE THE DRIVE THROUGH PORT B
```

```

6853
6854 043172 113760 001226 000010      MOV  PORTB,RPCS2(RO) ;SELECT PORT B
6855 043200 013737 001226 001236      MOV  PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
6856 043206 005060 000012      CLR  RPDS1(RO) ;WRITE RPDS1
6857 043212 013737 001224 001240      MOV  PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
6858
6859 ;:*****
6860 ;:FORCE AN ERROR
6861
6862 043220 012760 177777 000014      MOV  #-1,RPER1(RO) ;SET ERROR BITS
6863 043226 012760 000013 000000      MOV  #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
6864 043234 005037 001244      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6865 043240 016037 000000 001126      MOV  RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
6866 043246 012737 000000 001122      MOV  #RPCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6867 043254 060037 001122      ADD  RO,SBADR ;ADD RH11 BASE ADDRESS
6868 043260 012737 004012 001124      MOV  #4012,$GDDAT ;WHAT REGISTER SHOULD BE
6869 043266 013737 001126 001164      MOV  SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6870 043274 042737 173765 001164      BIC  #14012,$TMP0 ;SAVE SPECIFIED BITS
6871 043302 023737 001124 001164      CMP  $GDDAT,$TMP0 ;COMPARE THE BITS
6872 043310 001414      BEQ  66$ ;BR IF OK
6873 043312 013737 001126 001174      MOV  SBDDAT,$TMP4 ;COPY 'BAD DATA'
6874 043320 042737 004012 001174      BIC  #4012,$TMP4 ;CLEAR THE MASKED BITS
6875 043326 053737 001174 001124      BIS  $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6876 043334 104025      ERROR 25 ;TYPE MESSAGE 25
6877 043336 005137 001244      COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6878 043342 000240      NOP
6879 043344 005737 001244      TST  CKERR ;DID 'GO' BIT RESET ?
6880 043350 001002      BNE  .+6 ;BR IF NOT
6881 043352 000137 043412      JMP  1$ ;'GO' BIT RESET
6882 043356 012760 000040 000010      MOV  #CLR,RPCS2(RO) ;INIT THE RH11
6883 043364 113760 001226 000010      MOV  PORTB,RPCS2(RO) ;SELECT PORT B
6884 043372 013737 001226 001234      MOV  PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6885 043400 012760 000013 000000      MOV  #13,RPCS1(RO) ;RELEASE THE DRIVE THROUGH PORT B
6886 043406 000137 044126      JMP  2$ ;BYPASS THE REST OF THE TEST
6887
6888 ;:*****
6889 ;:VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
6890
6891 043412      1$:
6892 043412 113760 001224 000010      MOV  PORTA,RPCS2(RO) ;SELECT PORT A
6893 043420 013737 001224 001234      MOV  PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6894 043426 005037 001244      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6895 043432 016037 000012 001126      MOV  RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
6896 043440 012737 000012 001122      MOV  #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6897 043446 060037 001122      ADD  RO,SBADR ;ADD RH11 BASE ADDRESS
6898 043452 005037 001124      CLR  $GDDAT ;WHAT REGISTER SHOULD BE
6899 043456 023737 001124 001126      CMP  $GDDAT,SBDDAT ;IS THE REGISTER OK ?
6900 043464 001403      BEQ  68$ ;BR IF OK
6901 043466 104024      ERROR 24 ;TYPE MESSAGE 24
6902 043470 005137 001244      COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6903 043474 000240      NOP
6904 043476 113760 001226 000010      MOV  PORTB,RPCS2(RO) ;SELECT PORT B
6905 043504 013737 001226 001234      MOV  PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6906 043512 005037 001244      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6907 043516 016037 000014 001126      MOV  RPER1(RO),SBDDAT ;GET CONTENTS OF RPER1
6908 043524 012737 000014 001122      MOV  #RPER1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

J10

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 126
DZRJEA.CMB 02-NOV-76 18:40 T34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET

```
6909 043532 060037 001122          ADD    RO,$BDADR      ;ADD RH11 BASE ADDRESS
6910 043536 012737 177777 001124    MOV    #177777,$GDDAT ;WHAT REGISTER SHOULD BE
6911 043544 023737 001124 001126    CMP    $GDDAT,$BDDAT  ;IS THE REGISTER OK ?
6912 043552 001403          BEQ    70$            ;BR IF OK
6913 043554 104010          ERROR  10            ;REPORT THE ERROR
6914 043556 005137 001244          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
6915 043562 000240          70$:  NOP
6916
6917 ;*****
6918 ;CLEAR THE ERRORS THROUGH PORT B
6919
6920 043564 012760 000011 000000          MOV    #11,RPCS1(RO) ;ISSUE A DRIVE CLEAR
6921
6922 ;*****
6923
6924 ;RELEASE THE DRIVE FROM PORT B
6925
6926 043572 113760 001226 000010          MOV    PORTB,RPCS2(RO) ;SELECT PORT B
6927 043600 013737 001226 001234          MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6928 043606 012760 000013 000000          MOV    #13,RPCS1(RO) ;ISSLE RELEASE THROUGH PORT B
6929
6930 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6931
6932 043614 005037 001250          CLR    RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6933 043620 012737 000012 001122          MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6934 043626 060037 001122          ADD    RO,$BDADR     ;ADD THE I/O BASE ADDRESS
6935 043632 012737 011700 001124          MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6936 043640 113760 001224 000010          MOV    PORTA,RPCS2(RO) ;SELECT PORT A.
6937 043646 016037 000012 001170          MOV    RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6938 043654 013737 001170 001164          MOV    $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
6939 043662 042737 100100 001164          BIC    #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6940 043670 113760 001226 000010          MOV    PORTB,RPCS2(RO) ;SELECT PORT B.
6941 043676 016037 000012 001172          MOV    RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6942 043704 013737 001172 001166          MOV    $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
6943 043712 042737 100100 001166          BIC    #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6944 043720 023737 001164 001166          CMP    $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6945 043726 001006          BNE    72$           ;BR IF NOT
6946 043730 005737 001164          TST    $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6947 043734 001045          BNE    74$           ;BR IF NOT
6948 043736 104046          ERROR  46            ;REPORT DRIVE NOT IN NEUTRAL OR NCT SEIZED
6949 043740 000137 044124          JMP    76$           ;BYPASS THE REST OF THE CHECKS
6950 043744 013737 001170 001126 72$:  MOV    $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERKUR MESSAGE
6951 043752 013737 001226 001234          MOV    PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6952 043760 113760 001226 000010          MOV    PORTB,RPCS2(RO) ;SELECT PORT B.
6953 043766 005737 001164          TST    $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
6954 043772 001414          BEQ    73$           ;BR IF ZERO
6955 043774 013737 001224 001234          MOV    PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6956 044002 013737 001172 001126          MOV    $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
6957 044010 113760 001224 000010          MOV    PORTA,RPCS2(RO) ;SELECT PORT A.
6958 044016 005737 001166          TST    $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
6959 044022 001012          BNE    74$           ;BR IF NOT
6960 044024 012737 177777 001250 73$:  MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
6961 044032 012760 000011 000000          MOV    #11,RPCS1(RO) ;CLEAR THE DRIVE
6962 044040 012760 000013 000000          MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
6963 044046 104026          ERROR  26            ;TYPE ERROR MESSAGE 26
6964 044050 013737 001170 001126 74$:  MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
```


K10

MC-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 127
DZRJEA.CMB 02-NOV-76 18:40 T34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET

```
6965 044056 013737 001224 001234      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
6966 044064 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
6967 044072 001401                    BEQ      75$              ;BR IF OK FROM PORT A.
6968 044074 104007                    ERROR    7                ;REPORT ERROR
6969 04407E 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PCRT B.
6970 044104 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
6971 044112 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6972 044120 001401                    BEQ      76$              ;BR IF OK
6973 044122 104007                    ERROR    7                ;REPORT ERROR
6974 044124 000240 76$:  NOP
6975 044126 000004 2$:  SCOPE                      ;LOOP ?
```

```
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
```

```
*****
*TEST 35      TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
*
*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
*  B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 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.
*****
```

```
*****
*TEST35:
*
*  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
*  MOV      #35,$STSTNM     ;TEST NUMBER
*  MOV      #TEST35,$LPADR  ;LOAD LOOP ON TEST ADDRESS
*  MOV      #TEST35,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
*  MOV      #4,$TIMES       ;DO 4 ITERATIONS
*  MOV      #STACK,SP      ;LOAD THE STACK POINTER
*
*  ;CLEAR ATTENTION BITS FOR BOTH PORTS
*
*  MOV      PORTA,RPCS2(RO) ;SELECT PORT #A
*  CLR      RPDS1(RO)      ;SEIZE THE DRIVE
*  MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
*  MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
*  MOV      PORTB,RPCS2(RO) ;SELECT PORT #B
*  CLR      RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
*  MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
*  MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
*****
```

```
*****
*SEIZE THE DRIVE THROUGH PORT A
*****
```

L10

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 128
 DZRJEA.CMB 02-NOV-76 18:40 T35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

```

7021 044262 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
7022 044270 013737 001224 001236      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
7023 044276 005060 000012                CLR    RPDS1(RO) ;WRITE RPDS1
7024 044302 013737 001226 001240      MOV    PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
7025
7026 ;:*****
7027 ;WAIT 500 MS
7028
7029
7030 ;:*****
7031 ;START THE TIMER
7032
7033 044310 005037 001252                CLR    TIME ;CLEAR THE ELAPSED TIME COUNTER
7034 044314 012737 000764 001254      MOV    #500.,WATCH ;SET WATCH TO 500 MS
7035 044322 005737 001254      1$:   TST    WATCH ;WATCH EQUAL TO ZERO
7036 044326 001375                BNE    1$ ;BR IF NOT
7037
7038 ;:*****
7039 ;START THE TIMER
7040
7041 044330 005037 001252                CLR    TIME ;CLEAR THE ELAPSED TIME COUNTER
7042 044334 012737 003720 001254      MOV    #2000.,WATCH ;SET WATCH TO 2000 MS
7043
7044 ;:*****
7045 ;RETRIGGER THE TIMEOUT ONE-SHOT
7046
7047 044342 005760 000012                TST    RPDS1(RO) ;RETRIGGER THE ONE-SHOT
7048 044346 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
7049 044354 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7050 044362 005760 000012                TST    RPDS1(RO) ;WAIT FOR TIMEOUT
7051 044366 001004                BNE    3$ ;BR IF TIMEOUT OCCURRED
7052 044370 005737 001254                TST    WATCH ;WATCH EQUAL TO ZERO ?
7053 044374 001372                BNE    2$ ;BR IF NOT
7054 044376 104036                ERROR  36 ;NO TIMEOUT WITHIN 2 SECONDS
7055 044400 013737 001252 001272      3$:   MOV    TIME,1TIMES ;SAVE THE ELAPSED TIME VALUE
7056
7057 ;:*****
7058
7059 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7060
7061 044406 005037 001250                CLR    RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7062 044412 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7063 044420 060037 001122                ADD    RO,$BDADR ;ADD THE I/O BASE ADDRESS
7064 044424 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7065 044432 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
7066 044440 016037 000012 001170      MOV    RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7067 044446 013737 001170 001164      MOV    $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
7068 044454 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7069 044462 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
7070 044470 016037 000012 001172      MOV    RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7071 044476 013737 001172 001166      MOV    $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
7072 044504 042737 100100 001166      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7073 044512 023737 001164 001166      CMP    $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7074 044520 001006                BNE    66$ ;BR IF NOT
7075 044522 005737 001164                TST    $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7076 044526 001045                BNE    68$ ;BR IF NOT
  
```

M10

```

7077 044530 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7078 044532 000137 044716    JMP      70$      ;BYPASS THE REST OF THE CHECKS
7079 044536 013737 001170 001126 66$: MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7080 044544 013737 001226 001234    MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7081 044552 113760 001226 000010    MOVVB  PORTB,RPCS2(RO) ;SELECT PORT B.
7082 044560 005737 001164          TST     $TMP0     ;SEE IF STATUS EQ 0 FROM PCRT A.
7083 044564 001414          BEQ     67$      ;BR IF ZERO
7084 044566 013737 001224 001234    MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7085 044574 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7086 044602 113760 001224 000010    MOVVB  PORTA,RPCS2(RO) ;SELECT PORT A.
7087 044610 005737 001166          TST     $TMP1     ;SEE IF STATUS EQ ZERO FROM PORT B.
7088 044614 001012          BNE     68$      ;BR IF NOT
7089 044616 012737 177777 001250 67$: MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7090 044624 012760 000011 000000    MOV     #11,RPCS1(RO) ;CLEAR THE DRIVE
7091 044632 012760 000013 000000    MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
7092 044640 104022          ERROR 22          ;TYPE ERROR MESSAGE 22
7093 044642 013737 001170 001126 68$: MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7094 044650 013737 001224 001234    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
7095 044656 023737 001124 001170    CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
7096 044664 001401          BEQ     69$      ;BR IF OK FROM PORT A.
7097 044666 104007          ERROR 7           ;REPORT ERROR
7098 044670 013737 001172 001126 69$: MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7099 044676 013737 001226 001234    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
7100 044704 023737 001124 001172    CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
7101 044712 001401          BEQ     70$      ;BR IF OK
7102 044714 104007          ERROR 7           ;REPORT ERROR
7103 044716 000240          NOP

```

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

7108 044720 023737 001272 001260    CMP     TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7109 044726 003004          BGT     4$       ;BR IF GREATER
7110 044730 023737 001272 001262    CMP     TIMES,TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
7111 044736 002001          BGE     +4      ;BR IF NOT
7112 044740 104025          ERROR 25        ;REPORT THE ERROR
7113 044742 000004          SCOPE          ;LOOP ?

```

```

*****
;TEST 36 TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
;
;VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
;
; A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
;
; B. WAIT 500 MS AND WRITE 0'B INTO RPDS1 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.

```

```

7130 *****
7131 044744          TST36: TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7132 044744 005737 001274

```

N10

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 130
 DZRJEA.CMB 02-NOV-76 18:40 T36 TEST TIMEOUT RETRIGGER THROUGH PORT 'B'

```

7133 044750 001406          BEQ     2$          ;BR IF NOT
7134 044752 100002          BPL     1$          ;BR IF JUST ENTERED TEST
7135 044754 000137 002602     JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
7136 044760 012737 177777 001274 1$: MOV     #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
7137 044766 112737 000036 001102 2$: MOV    #36,$TSTNM ;TEST NUMBER
7138 044774 012737 045016 001106     MOV    #TEST36,$LPADR ;LOAD LOOP ON TEST ADDRESS
7139 045002 012737 045016 001110     MOV    #TEST36,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7140 045010 012737 000004 001176     MOV    #4,$TIMES     ;DO 4 ITERATIONS
7141 045016 012706 001100 TEST36: MOV   #STACK,SP ;LOAD THE STACK POINTER
7142
7143          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7144
7145 045022 113760 001224 000010     MOV    PORTA,RPDS2(RO) ;SELECT PORT #A
7146 045030 005060 000012          CLR    RPDS1(RO)      ;SEIZE THE DRIVE
7147 045034 012760 000011 000000     MOV    #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
7148 045042 012757 000013 000000     MOV    #13,RPDS1(RO) ;RELEASE THE DRIVE
7149 045050 113760 001226 000010     MOV    PORTB,RPDS2(RO) ;SELECT PORT #B
7150 045056 005060 000012          CLR    RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
7151 045062 012760 000011 000000     MOV    #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
7152 045070 012760 000013 000000     MOV    #13,RPDS1(RO) ;RELEASE THE DRIVE
7153
7154          ;*****
7155
7156          ;SEIZE THE DRIVE THROUGH PORT B
7157
7158 045076 113760 001226 000010     MOV    PORTB,RPDS2(RO) ;SELECT PORT B
7159 045104 013737 001226 001236     MOV    PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
7160 045112 005060 000012          CLR    RPDS1(RO)      ;WRITE RPDS1
7161 045116 013737 001224 001240     MOV    PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
7162
7163          ;*****
7164          ;WAIT 500 MS
7165
7166          ;*****
7167          ;START THE TIMER
7168
7169
7170 045124 005037 001252          CLR    TIME          ;CLEAR THE ELAPSED TIME COUNTER
7171 045130 012737 000764 001254     MOV    #500.,WATCH  ;SET WATCH TO 500 MS
7172 045136 005737 001254 1$: TST    WATCH        ;WATCH EQUAL TO ZERO
7173 045142 001375          BNE    1$          ;BR IF NOT
7174
7175          ;*****
7176          ;START THE TIMER
7177
7178 045144 005037 001252          CLR    TIME          ;CLEAR THE ELAPSED TIME COUNTER
7179 045150 012737 003720 001254     MOV    #2000.,WATCH ;SET WATCH TO 2000 MS
7180
7181          ;*****
7182          ;RETRIGGER THE TIMEOUT ONE-SHOT
7183
7184 045156 005760 000012          TST    RPDS1(RO)     ;RETRIGGER THE ONE-SHOT
7185 045162 113760 001224 000010     MOV    PORTA,RPDS2(RO) ;SELECT PORT A
7186 045170 013737 001224 001234     MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7187 045176 005760 000012 2$: TST    RPDS1(RO)     ;WAIT FOR TIMEOUT
7188 045202 001004          BNE    3$          ;BR IF TIMEOUT OCCURRED
  
```

B11

```

7189 045204 005737 001254          TST      WATCH          ;WATCH EQUAL TO ZERO ?
7190 045210 001372          BNE      25              ;BR IF NOT
7191 045212 104036          ERROR    36              ;NO TIMEOUT WITHIN 2 SECONDS
7192 045214 013737 001252 001272 35:    MOV      TIME,TIMES      ;SAVE THE ELAPSED TIME VALLE
7193
7194
7195
7196
7197
7198
7199
7200 045222 005937 001250          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7201 045226 012737 000012 001122    MOV      #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7202 045234 060037 001122          ADD      RO,$BDADR      ;ADD THE I/O BASE ADDRESS
7203 045240 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7204 045246 113760 001224 000010    MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
7205 045254 016037 000012 001170    MOV      RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7206 045262 013737 001170 001164    MOV      STMP2,STMP0     ;COPY IT INTO 'STMP0'
7207 045270 042737 100100 001164    BIC      #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7208 045276 113760 001226 000010    MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
7209 045304 016037 000012 001172    MOV      RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7210 045312 013737 001172 001166    MOV      STMP3,STMP1    ;COPY IT INTO 'STMP1'
7211 045320 042737 100100 001166    BIC      #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7212 045326 023737 001164 001166    CMP      STMP0,STMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7213 045334 001006          BNE      66$            ;BR IF NOT
7214 045336 005737 001164          TST      STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7215 045342 001045          BNE      68$            ;BR IF NOT
7216 045344 104046          ERROR    46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7217 045346 000137 045532          JMP      70$            ;BYPASS THE REST OF THE CHECKS
7218 045352 013737 001170 001126 66$:    MOV      STMP2,$BDADR   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7219 045360 013737 001226 001234    MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7220 045366 113760 001226 000010    MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
7221 045374 005737 001164          TST      STMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
7222 045400 001414          BEQ      67$            ;BR IF ZERO
7223 045402 013737 001224 001234    MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7224 045410 013737 001172 001126    MOV      STMP3,$BDADR   ;'BAD DATA' FOR ERROR TYPE OUT
7225 045416 113760 001224 000010    MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
7226 045424 005737 001166          TST      STMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
7227 045430 001012          BNE      68$            ;BR IF NOT
7228 045432 012737 177777 001250 67$:    MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
7229 045440 012760 000011 000000    MOV      #11,RPCS1(RO) ;CLEAR THE DRIVE
7230 045446 012760 000013 000000    MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
7231 045454 104022          ERROR    22              ;TYPE ERROR MESSAGE 22
7232 045456 013737 001170 001126 68$:    MOV      STMP2,$BDADR   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7233 045464 013737 001224 001234    MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
7234 045472 023737 001124 001170    CMP      $GDDAT,STMP2  ;ALL BITS OK ?
7235 045500 001401          BEQ      69$            ;BR IF OK FROM PORT A.
7236 045502 104007          ERROR    7              ;REPORT ERROR
7237 045504 013737 001172 001126 69$:    MOV      STMP3,$BDADR   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7238 045512 013737 001226 001234    MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
7239 045520 023737 001124 001172    CMP      $GDDAT,STMP3  ;SEE IF READ OK FROM PORT B.
7240 045526 001401          BEQ      70$            ;BR IF OK
7241 045530 104007          ERROR    7              ;REPORT ERROR
7242 045532 000240          NOP
7243
7244

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

C11

```

7245 045534 023737 001272 001266      CMP      TIMES,TIMEBP      ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7246 045542 003004                      BGT      45                ;BR IF GREATER
7247 045544 023737 001272 001270      CMP      TIMES,TIMEBM      ;MEASURED TIME LESS THAN -25% TOLERANCE
7248 045552 002001                      BGE      +4                ;BR IF NOT
7249 045554 104025                      ERROR    25                ;REPORT THE ERROR
7250 045556 000004                      SCOPE                                ;LOOP ?

```

:TEST 37 TEST PORT 'A' ATTENTION AFTER A COMMAND

- * TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A COMMAND.
- * A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
- * B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME 'I'). 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.

```

7271 045560                      TST37:
7272 045560 005737 001274      TST      KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
7273 045564 001406                      BEQ      25                ;BR IF NOT
7274 045566 100002                      BPL      15                ;BR IF JUST ENTERED TEST
7275 045570 000137 002602      JMP      EXEC              ;RETURN & GET NEXT TEST NUMBER
7276 045574 012737 177777 001274 15:  MOV      #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
7277 045602 112737 000037 001102 25:  MOVVB   #37,$STNM         ;TEST NUMBER
7278 045610 012737 045632 001106      MOV      #TEST37,$LPADR    ;LOAD LOOP ON TEST ADDRESS
7279 045616 012737 045632 001110      MOV      #TEST37,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
7280 045624 012737 000004 001176      MOV      #4,$TIMES         ;DO 4 ITERATIONS
7281 045632 012706 001100      TEST37: MOV      #STACK,SP    ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

7285 045636 113760 001224 000010      MOVVB   PORTA,RPCS2(RO)    ;SELECT PORT #A
7286 045644 005060 000012                      CLR      RPDS1(RO)         ;SEIZE THE DRIVE
7287 045650 012760 000011 000000      MOV      #11,RPCS1(RO)     ;ISSUE DRIVE CLEAR
7288 045656 012760 000013 000000      MOV      #13,RPCS1(RO)     ;RELEASE THE DRIVE
7289 045664 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)    ;SELECT PORT #B
7290 045672 005060 000012                      CLR      RPDS1(RO)         ;SEIZE THE DRIVE THROUGH PORT 'B'
7291 045676 012760 000011 000000      MOV      #11,RPCS1(RO)     ;ISSUE DRIVE CLEAR
7292 045704 012760 000013 000000      MOV      #13,RPCS1(RO)     ;RELEASE THE DRIVE
7293 045712 113760 001224 000010      MOVVB   PORTA,RPCS2(RO)    ;SELECT PORT A
7294 045720 013737 001224 001234      MOV      PORTA,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7295 045726 013737 001224 001236      MOV      PORTA,SEIZPT      ;'SEIZED' PORT ADDRESS

```

:DO A RECALIBRATE THROUGH PORT A

```

7300 045734 012760 000007 000000      MOV      #7,RPCS1(RO)      ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A

```

D11

7301
7302
7303
7304
7305 045742 032760 000200 000012
7306 045750 001774
7307
7308
7309
7310
7311 045752 005037 001244
7312 045756 016037 000012 001126
7313 045764 012737 000012 001122
7314 045772 060037 001122
7315 045776 012737 100000 001124
7316 046004 013737 001126 001164
7317 046012 042737 077777 001164
7318 046020 023737 001124 001164
7319 046026 001414
7320 046030 013737 001126 001174
7321 046036 042737 100000 001174
7322 046044 053737 001174 001124
7323 046052 104032
7324 046054 005137 001244
7325 046060 000240
7326 046062 113760 001226 000010
7327 046070 013737 001226 001234
7328
7329
7330
7331
7332 046076 005037 001244
7333 046102 016037 000012 001126
7334 046110 012737 000012 001122
7335 046116 060037 001122
7336 046122 005037 001124
7337 046126 013737 001126 001164
7338 046134 042737 077777 001164
7339 046142 023737 001124 001164
7340 046150 001414
7341 046152 013737 001126 001174
7342 046160 042737 100000 001174
7343 046166 053737 001174 001124
7344 046174 104032
7345 046176 005137 001244
7346 046202 000240
7347
7348
7349
7350
7351
7352 046204 113760 001224 000010
7353 046212 013737 001224 001234
7354 046220 012760 000013 000000
7355
7356

;WAIT FOR DRIVE TO FINISH

BIT #DRY,RPDS1(RO) ;WAIT FOR DRIVE TO FINISH
BEQ .-6 ;BR IF NOT FINISHED

;CONFIRM THAT ATTENTION IS SET FOR PORT A

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
MOV #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #ICATA,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 64\$;BR IF OK
MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 32 ;TYPE MESSAGE 32
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
64\$: NOP
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
CLR SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #ICATA,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 66\$;BR IF OK
MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 32 ;TYPE MESSAGE 32
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
66\$: NOP

;RELEASE THE DRIVE FROM PORT A

MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

E11

```

7357
7358 046226 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7359 046232 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7360 046240 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
7361 046244 012737 011700 001124 MOV #M0L!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7362 046252 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
7363 046260 016037 000012 001170 MOV RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7364 046266 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
7365 046274 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7366 046302 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
7367 046310 016037 000012 001172 MOV RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7368 046316 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7369 046324 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7370 046332 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7371 046340 001006 BNE 68$ ;BR IF NOT
7372 046342 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7373 046346 001045 BNE 70$ ;BR IF NOT
7374 046350 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7375 046352 000137 046536 JMP 72$ ;BYPASS THE REST OF THE CHECKS
7376 046356 013737 001170 001126 69$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7377 046364 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7378 046372 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
7379 046400 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7380 046404 001414 BEQ 69$ ;BR IF ZERO
7381 046406 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7382 046414 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7383 046422 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
7384 046430 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7385 046434 001012 BNE 70$ ;BR IF NOT
7386 046436 012737 177777 001250 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7387 046444 012760 000011 000000 MOV #11,RPCS1(R0) ;CLEAR THE DRIVE
7388 046452 012760 000013 000000 MOV #13,RPCS1(R0) ;RELEASE THE DRIVE
7389 046460 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
7390 046462 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7391 046470 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7392 046476 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
7393 046504 001401 BEQ 71$ ;BR IF OK FROM PORT A.
7394 046506 104007 ERROR 7 ;REPORT ERROR
7395 046510 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7396 046516 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
7397 046524 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
7398 046532 001401 BEQ 72$ ;BR IF OK
7399 046534 104007 ERROR 7 ;REPORT ERROR
7400 046536 000240 72$: NOP
7401 046540 000004 SCOPE ;LOOP ?

```

```

7402
7403 *****
7404 *TEST 40 TEST PORT 'B' ATTENTION AFTER A COMMAND
7405 *
7406 *TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
7407 * COMMAND.
7408 *
7409 * A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
7410 *
7411 * B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
7412 * '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND

```


F11

```

7413          : *      THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
7414          : *
7415          : * C.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
7416          : *      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7417          : *
7418          : *****
7419          : TST40:
7419 046542          TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7420 046542 005737 001274      BEQ      25          ;BR IF NOT
7421 046546 001406          BPL      15          ;BR IF JUST ENTERED TEST
7422 046550 100002          JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
7423 046552 000137 002602      MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
7424 046556 012737 177777 001274 15:      MOVVB   #40,$STSTNM ;TEST NUMBER
7425 046564 112737 000040 001102 25:      MOV      #TEST40,$LPADR ;LOAD LOOP ON TEST ADDRESS
7426 046572 012737 046614 001106          MOV      #TEST40,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7427 046500 012737 046614 001110          MOV      #4,$TIMES    ;DO 4 ITERATIONS
7428 046606 012737 000004 001176          TEST40: MOV     #STACK,$SP ;LOAD THE STACK POINTER
7429 046614 012706 001100
7430
7431          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7432
7433 046620 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT #A
7434 046626 005060 000012          CLR      RPDS1(RO)     ;SEIZE THE DRIVE
7435 046632 012760 000011 000000      MOV      #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
7436 046640 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
7437 046646 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT #B
7438 046654 005060 000012          CLR      RPDS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
7439 046660 012760 000011 000000      MOV      #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
7440 046666 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
7441 046674 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
7442 046702 013737 001226 001234      MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7443 046710 013737 001226 001236      MOV      PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
7444
7445          : *****
7446          : ;DO A RECALIBRATE THROUGH PORT B
7447
7448 046716 012760 000007 000000      MOV      #7,RPCS1(RO) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B
7449
7450          : *****
7451          : ;WAIT FOR DRIVE TO FINISH
7452
7453 046724 032760 000200 000012      BIT      #DRY,RPDS1(RO) ;WAIT FOR DRIVE TO FINISH
7454 046732 001774          BEQ      -6          ;BR IF NOT FINISHED
7455
7456          : *****
7457          : ;CONFIRM THAT ATTENTION IS SET FOR PORT B
7458
7459 046734 005037 001244          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
7460 046740 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
7461 046746 012737 000012 001122      MOV      #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7462 046754 060037 001122          ADD      RO,$B0ADR    ;ADD RH11 BASE ADDRESS
7463 046760 012737 100000 001124      MOV      #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
7464 046766 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7465 046774 042737 077777 001164      BIC      #CATA,$TMP0  ;SAVE SPECIFIED BITS
7466 047002 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
7467 047010 001414          BEQ      64$         ;BR IF OK
7468 047012 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
  
```

G11

MD-11-DZJTE-A, DUAL CONTR. R LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 136
 DZJTEA.CMB 02-NOV-76 18:40 T40 TEST PORT 'B' ATTENTION AFTER A COMMAND

```

7469 047020 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
7470 047026 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
7471 047034 104032                ERROR    32             ;TYPE MESSAGE 32
7472 047036 005137 001244                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
7473 047042 000240                64$:    NOP
7474 047044 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
7475 047052 013737 001224 001234      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7476
7477
7478
7479
7480 047060 005037 001244                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
7481 047064 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
7482 047072 012737 000012 001122      MOV      #RPDS1,$BADDR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7483 047100 060037 001122                ADD      RO,$BADDR     ;ADD RHI1 BASE ADDRESS
7484 047104 005037 001124                CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
7485 047110 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
7486 047116 042737 077777 001164      BIC      #↑CATA,$TMP0   ;SAVE SPECIFIED BITS
7487 047124 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
7488 047132 001414                BEQ      66$           ;BR IF OK
7489 047134 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
7490 047142 042737 100000 001174      BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
7491 047150 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
7492 047156 104032                ERROR    32             ;TYPE MESSAGE 32
7493 047160 005137 001244                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
7494 047164 000240                66$:    NOP
7495
7496
7497
7498
7499
7500 047166 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
7501 047174 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7502 047202 012760 000013 000000      MOV      #13,RPDS1(RO)  ;ISSUE RELEASE THROUGH PORT B
7503
7504
7505
7506 047210 005037 001250                CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
7507 047214 012737 000012 001122      MOV      #RPDS1,$BADDR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7508 047222 060037 001122                ADD      RO,$BADDR     ;ADD THE I/O BASE ADDRESS
7509 047226 012737 011700 001124      MOV      #10L!FGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7510 047234 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A.
7511 047242 016037 000012 001170      MOV      RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7512 047250 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
7513 047256 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7514 047264 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B.
7515 047272 016037 000012 001172      MOV      RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7516 047300 013737 001172 001166      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
7517 047306 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7518 047314 023737 001164 001166      CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7519 047322 001006                BNE      68$           ;BR IF NOT
7520 047324 005737 001164                TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7521 047330 001045                BNE      70$           ;BR IF NOT
7522 047332 104046                ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7523 047334 000137 047520                JMP      72$           ;BYPASS THE REST OF THE CHECKS
7524 047340 013737 001170 001126 68$:    MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE

```

H11

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 137
DZRJEA.CMB 02-NOV-76 18:40 T40 TEST PORT 'B' ATTENTION AFTER A COMMAND

7525	047346	013737	001226	001234	MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
7526	047354	113760	001226	000010	MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.	
7527	047362	005737	001164		TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.	
7528	047366	001414			BEQ	69\$;BR IF ZERO	
7529	047370	013737	001224	001234	MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
7530	047376	013737	001172	001126	MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT	
7531	047404	113760	001224	000010	MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.	
7532	047412	005737	001166		TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.	
7533	047416	001012			BNE	70\$;BR IF NOT	
7534	047420	012737	177777	001250	69\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
7535	047426	012760	000011	000000	MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE	
7536	047434	012760	000013	000000	MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE	
7537	047442	104026			ERROR	26	;TYPE ERROR MESSAGE 26	
7538	047444	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7539	047452	013737	001224	001234	MOV	PORTA,PTNBR	;CHANGE PORT NUMBER	
7540	047460	023737	001124	001170	CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?	
7541	047466	001401			BEQ	71\$;BR IF OK FROM PORT A.	
7542	047470	104007			ERROR	7	;REPORT ERROR	
7543	047472	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7544	047500	013737	001226	001234	MOV	PORTB,PTNBR	;CHANGE PORT NUMBER	
7545	047506	023737	001124	001172	CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.	
7546	047514	001401			BEQ	72\$;BR IF OK	
7547	047516	104007			ERROR	7	;REPORT ERROR	
7548	047520	000240			72\$:	NOP		
7549	047522	000004			SCOPE		;LOOP ?	

7550
7551
7552
7553
7554
7555
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

```
*****  
*TEST 41 TEST PORT INTERACTION FROM PORT 'A'  
*  
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.  
*  
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.  
*  
* B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  
*  
* C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT  
* 'B' SEES 0'S FROM EACH OF THESE REGISTERS.  
*  
* D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  
*  
* E. WRITE 1'S INTO RPER1, RPER2, & RPER3 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.  
*  
*****  
TST41: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?  
REQ 26 ;BR IF NOT  
BPL 1$ ;BR IF JUST ENTERED TEST
```

047524							
047524	005737	001274					
047530	001406						
047532	!00002						

7581	047534	000137	002602		JMP	EYEC	;RETURN & GET NEXT TEST NUMBER
7582	047540	012737	177777	001274	15:	MOV	#-1,KYBCTL ;SET SINGLE TEST INDICATOR
7583	047546	112737	000041	001102	25:	MOVB	#41,\$STSNM ;TEST NUMBER
7584	047554	012737	047576	001106		MOV	#TEST41,\$LPADR ;LOAD LOOP ON TEST ADDRESS
7585	047562	012737	047576	001110		MOV	#TEST41,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
7586	047570	012737	007640	001176		MOV	#4000,\$TIMES ;DO 4000. ITERATIONS
7587	047576	012706	001100		TEST41:	MOV	#STACK,\$P ;LOAD THE STACK POINTER
7588							
7589							
7590							;CLEAR ATTENTION BITS FOR BOTH PORTS
7591	047602	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT #A
7592	047610	005060	000012			CLR	RPDS1(RO) ;SEIZE THE DRIVE
7593	047614	012760	000011	000000		MOV	#11,RPCS1(RO) ;ISSUE DRIVE CLEAR
7594	047622	012760	000013	000000		MOV	#13,RPCS1(RO) ;RELEASE THE DRIVE
7595	047630	113760	001226	000010		MOVB	PORTB,RPCS2(RO) ;SELECT PORT #B
7596	047636	005060	000012			CLR	RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
7597	047642	012760	000011	000000		MOV	#11,RPCS1(RO) ;ISSUE DRIVE CLEAR
7598	047650	012760	000013	000000		MOV	#13,RPCS1(RO) ;RELEASE THE DRIVE
7599							
7600							;SEIZE THE DRIVE THROUGH PORT A
7601							
7602	047656	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
7603	047664	013737	001224	001236		MOV	PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
7604	047672	005060	000012			CLR	RPDS1(RO) ;WRITE RPDS1
7605	047676	013737	001226	001240		MOV	PORTB,OPPR ;'OPPOSITE' PORT ADDRESS
7606	047704	012760	177777	000014		MOV	#-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT A
7607	047712	012760	177777	000040		MOV	#-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT A
7608	047720	012760	177777	000042		MOV	#-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT A
7609	047726	113760	001226	000010		MOVB	PORTB,RPCS2(RO) ;SELECT PORT B
7610	047734	013737	001226	001234		MOV	PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7611	047742	004737	050600			JSR	PC,TST41B ;CHECK THE REGISTERS THROUGH PORT B
7612	047746	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
7613	047754	013737	001224	001234		MOV	PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7614	047762	005060	000042			CLR	RPER3(RO) ;CLEAR RPER3 ON PORT A
7615	047766	005060	000040			CLR	RPER2(RO) ;CLEAR RPER2 ON PORT A
7616	047772	005060	000014			CLR	RPER1(RO) ;CLEAR RPER1 ON PORT A
7617	047776	013760	001232	000016		MOV	ASR1,RPAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT A
7618	050004	113760	001226	000010		MOVB	PORTB,RPCS2(RO) ;SELECT PORT B
7619	050012	013737	001226	001234		MOV	PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7620	050020	012760	177777	000014		MOV	#-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT B
7621	050026	012760	177777	000040		MOV	#-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT B
7622	050034	012760	177777	000042		MOV	#-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT B
7623	050042	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
7624	050050	013737	001224	001234		MOV	PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7625	050056	004737	050600			JSR	PC,TST41B ;CHECK THE REGISTERS THROUGH PORT A
7626							
7627							;RELEASE THE DRIVE FROM PORT A
7628							
7629	050062	113760	001224	000010		MOVB	PORTA,RPCS2(RO) ;SELECT PORT A
7630	050070	013737	001224	001234		MOV	PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7631	050076	012760	000013	000000		MOV	#13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7632							
7633							;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
7634							
7635	050104	005037	001250			CLR	RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
7636	050110	012737	111700	001124		MOV	#ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT

J11

7637	050116	012737	000012	001122		MOV	#RPDS1,\$BDADR	;REGISTER ADDRESS INCREMENT
7638	050124	060037	001122			ADD	RO,\$BDADR	;REGISTER BASE ADDRESS FOR TYPEOUT
7639	050130	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
7640	050136	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7641	050144	016037	000012	001164		MOV	RPDS1(RO),\$TMP0	;READ STATUS REGISTER FROM PORT B
7642	050152	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
7643	050160	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7644	050166	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;DRIVE STATUS FROM PORT A
7645	050174	001404				BEQ	66\$;BR IF STATUS FROM PORT A ZERO
7646	050176	005737	001164			TST	\$TMP0	;IS STATUS FROM PORT B ZERO ?
7647	050202	001401				BEQ	66\$;BR IF ZERO
7648	050204	104031				ERROR	31	;REPORT DRIVE IN NEUTRAL
7649	050206	013737	001164	001126	66\$:	MOV	\$TMP0,\$BDDAT	;CHECK STATUS FROM PORT B
7650	050214	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT ADDRESS FOR TYPEOUT
7651	050222	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;COMPARE WITH CONSTANT
7652	050230	001401				BEQ	67\$;BR IF OK
7653	050232	104027				ERROR	27	;REPORT REGISTER ERROR
7654	050234	000240			67\$:	NOP		
7655								
7656								;RELEASE THE DRIVE FROM PORT B
7657								
7658	050236	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
7659	050244	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7660	050252	012760	000013	000000		MOV	#13,RPCS1(RO)	;ISSUE RELEASE THROUGH PORT B
7661								
7662								;VERIFY THAT THE DRIVE IS IN NEUTRAL
7663								
7664	050260	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
7665	050264	012737	000012	001122		MOV	#RPDS1,\$BDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7666	050272	060037	001122			ADD	RO,\$BCADR	;ADD THE I/O BASE ADDRESS
7667	050276	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!	\$VV,\$GDDAT ;COMPARISON CONSTANT
7668	050304	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
7669	050312	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
7670	050320	013737	001170	001164		MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
7671	050326	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7672	050334	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
7673	050342	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
7674	050350	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
7675	050356	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7676	050364	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7677	050372	001006				BNE	68\$;BR IF NOT
7678	050374	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7679	050400	001045				BNE	70\$;BR IF NOT
7680	050402	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7681	050404	000137	050570			JMP	72\$;BYPASS THE REST OF THE CHECKS
7682	050410	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7683	050416	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7684	050424	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
7685	050432	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
7686	050436	001414				BEQ	69\$;BR IF ZERO
7687	050440	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7688	050446	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
7689	050454	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
7690	050462	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
7691	050466	001012				BNE	70\$;BR IF NOT
7692	050470	012737	177777	001250	69\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR

K11

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-75 19:42 PAGE 140
 DZRJEA.CMB 02-NOV-75 18:40 T41 TEST PORT INTERACTION FROM PORT 'A'

```

7693 050476 012760 000011 000000      MOV      #11,RPCS1(RO)      ;CLEAR THE DRIVE
7694 050504 012760 000013 000000      MOV      #13,RPCS1(RO)      ;RELEASE THE DRIVE
7695 050512 104026                ERROR    26                  ;TYPE ERROR MESSAGE 26
7696 050514 013737 001170 001126 70$:      MOV      $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7697 050522 013737 001224 001234      MOV      PORTA,PTNBR        ;CHANGE PORT NUMBER
7698 050530 023737 001124 001170      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
7699 050536 001401                BEQ      71$                ;BR IF OK FROM PORT A.
7700 050540 104007                ERROR    7                  ;REPORT ERROR
7701 050542 013737 001172 001126 71$:      MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7702 050550 013737 001226 001234      MOV      PORTB,PTNBR        ;CHANGE PORT NUMBER
7703 050556 023737 001124 001172      CMP      $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
7704 050564 001401                BEQ      72$                ;BR IF OK
7705 050566 104007                ERROR    7                  ;REPORT ERROR
7706 050570 000240                NOP
7707 050572 000004                SCOPE
7708 050574 000137 051132      JMP      TST42              ;LOOP ?
7709
7710
7711
7712 050600                TST41B:
7713 050600 005037 001244                CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
7714 050604 016037 000014 001126      MOV      RPER1(RO), $BDDAT  ;GET CONTENTS OF RPER1
7715 050612 012737 000014 001122      MOV      #RPER1,$BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7716 050620 060037 001122                ADD     RO,$BDADR          ;ADD RH11 BASE ADDRESS
7717 050624 005037 001124                CLR      $GDDAT            ;WHAT REGISTER SHOULD BE
7718 050630 023737 001124 001126      CMP      $GDDAT,$BDDAT     ;IS THE REGISTER OK ?
7719 050636 001403                BEQ      64$                ;BR IF OK
7720 050640 104006                ERROR    6                  ;TYPE MESSAGE 6
7721 050642 005137 001244                COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
7722 050646 016037 000000 001126 64$:      MOV      RPCS1(RO), $BDDAT  ;GET THE CONTENTS OF RHCS1
7723 050654 012737 000000 001122      MOV      #RPCS1,$BDADR     ;FORM ADDRESS OF REGISTER
7724 050662 060037 001122                ADD     RO,$BDADR          ;ADDRESS BASE
7725 050666 032737 020000 001126      BIT     #MCPE,$BDDAT       ;IS 'MCPE' SET ?
7726 050674 001404                BEQ      65$                ;BR IF NOT
7727 050676 104011                ERROR    11                 ;REPORT THE ERROR
7728 050700 012760 040000 000000      MOV      #TRE,RPCS1(RO)    ;CLEAR 'MCPE'
7729 050706 000240                NOP
7730 050710 005037 001244                CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
7731 050714 016037 000040 001126      MOV      RPER2(RO), $BDDAT  ;GET CONTENTS OF RPER2
7732 050722 012737 000040 001122      MOV      #RPER2,$BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7733 050730 060037 001122                ADD     RO,$BDADR          ;ADD RH11 BASE ADDRESS
7734 050734 005037 001124                CLR      $GDDAT            ;WHAT REGISTER SHOULD BE
7735 050740 023737 001124 001126      CMP      $GDDAT,$BDDAT     ;IS THE REGISTER OK ?
7736 050746 001403                BEQ      66$                ;BR IF OK
7737 050750 104006                ERROR    6                  ;TYPE MESSAGE 6
7738 050752 005137 001244                COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
7739 050756 016037 000000 001126 66$:      MOV      RPCS1(RO), $BDDAT  ;GET THE CONTENTS OF RHCS1
7740 050764 012737 000000 001122      MOV      #RPCS1,$BDADR     ;FORM ADDRESS OF REGISTER
7741 050772 060037 001122                ADD     RO,$BDADR          ;ADDRESS BASE
7742 050776 032737 020000 001126      BIT     #MCPE,$BDDAT       ;IS 'MCPE' SET ?
7743 051004 001404                BEQ      67$                ;BR IF NOT
7744 051006 104011                ERROR    11                 ;REPORT THE ERROR
7745 051010 012760 040000 000000      MOV      #TRE,RPCS1(RO)    ;CLEAR 'MCPE'
7746 051016 000240                NOP
7747 051020 005037 001244                CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
7748 051024 016037 000042 001126      MOV      RPER3(RO), $BDDAT  ;GET CONTENTS OF RPER3

```

L11

```

7749 051032 012737 000042 001122      MOV      #RPER3,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7750 051040 060037 001122      ADD      RO,$BDADR    ;ADD RH11 BASE ADDRESS
7751 051044 005037 001124      CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
7752 051050 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7753 051056 001403      BEQ      68$         ;BR IF OK
7754 051060 104006      ERROR    6          ;TYPE MESSAGE 6
7755 051062 005137 001244      COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
7756 051066 016037 000000 001126 68$:  MOV      RPCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
7757 051074 012737 000000 001122      MOV      #RPCS1,$BDADR ;FORM ADDRESS OF REGISTER
7758 051102 060037 001122      ADD      RO,$BDADR    ;ADDRESS BASE
7759 051106 032737 02C000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET ?
7760 051114 001404      BEQ      69$         ;BR IF NOT
7761 051116 104011      ERROR    11         ;REPORT THE ERROR
7762 051120 012760 040000 000000      MOV      #TRE,RPCS1(RO) ;CLEAR 'MCPE'
7763 051126 000240 69$:  NOP
7764 051130 000207      RTS      PC          ;RETURN
  
```

```

*****
*TEST 42      TEST PORT INTERACTION FROM PORT 'B'
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
*  B.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
*
*  C.  READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  VERIFY THAT PORT
*      'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  D.  CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
*
*  E.  WRITE 1'S INTO RPER1, RPER2, & RPER3 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.
*
*****
  
```

```

7791 051132      TST42:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
7792 051132 005737 001274      BEQ      2$          ;BR IF NOT
7793 051136 001406      BPL      1$          ;BR IF JUST ENTERED TEST
7794 051140 100002      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
7795 051142 000137 002602      MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
7796 051146 012737 177777 001274 1$:  MOVVB   #42,$TSTNM   ;TEST NUMBER
7797 051154 112737 000042 001102 2$:  MOV      #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
7798 051162 012737 051204 001106      MOV      #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7799 051170 012737 051204 001110      MOV      #4000,$TIMES ;DO 4000. ITERATIONS
7800 051176 012737 007640 001176      MOV      #STACK,$SP  ;LOAD THE STACK POINTER
7801 051204 012706 001100      TEST42: MOV
7802
7803      ;CLEAR ATTENTION BITS FOR BOTH PORTS
7804
  
```

M11

MO-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 142
 DZRJEA.CMS 02-NOV-76 18:40 T42 TEST PORT INTERACTION FROM PORT 'B'

7805	051210	113760	001224	000010	MOV B	PORTA, RPCS2(RO)	; SELECT PORT #A
7806	051216	005060	000012		CLR	RPDS1(RO)	; SEIZE THE DRIVE
7807	051222	012760	000011	000000	MOV	#11, RPCS1(RO)	; ISSUE DRIVE CLEAR
7808	051230	012760	000013	000000	MOV	#13, RPCS1(RO)	; RELEASE THE DRIVE
7809	051236	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT #B
7810	051244	005060	000012		CLR	RPDS1(RO)	; SEIZE THE DRIVE THROUGH PORT 'B'
7811	051250	012760	000011	000000	MOV	#11, RPCS1(RO)	; ISSUE DRIVE CLEAR
7812	051256	012760	000013	000000	MOV	#13, RPCS1(RO)	; RELEASE THE DRIVE
7813							
7814							; SEIZE THE DRIVE THROUGH PORT B
7815							
7816	051264	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT B
7817	051272	013737	001226	001236	MOV	PORTB, SEIZPT	; STORE SEIZING PORT'S ADDRESS
7818	051300	005060	000012		CLR	RPDS1(RO)	; WRITE RPDS1
7819	051304	013737	001224	001240	MOV	PORTA, OPPRT	; 'OPPOSITE' PORT ADDRESS
7820	051312	012760	177777	000014	MOV	#-1, RPER1(RO)	; LOAD 1'S INTO RPER1 THROUGH PORT B
7821	051320	012760	177777	000040	MOV	#-1, RPER2(RO)	; LOAD 1'S INTO RPER2 THROUGH PORT B
7822	051326	012760	177777	000042	MOV	#-1, RPER3(RO)	; LOAD 1'S INTO RPER3 THROUGH PORT B
7823	051334	113760	001224	000010	MOV B	PORTA, RPCS2(RO)	; SELECT PORT A
7824	051342	013737	001224	001234	MOV	PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7825	051350	004737	052206		JSR	PC, TST42B	; CHECK THE REGISTERS THROUGH PORT A
7826	051354	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT B
7827	051362	013737	001226	001234	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7828	051370	005060	000042		CLR	RPER3(RO)	; CLEAR RPER3 ON PORT B
7829	051374	005060	000040		CLR	RPER2(RO)	; CLEAR RPER2 ON PORT B
7830	051400	005060	000014		CLR	RPER1(RO)	; CLEAR RPER1 ON PORT B
7831	051404	013760	001232	000016	MOV	ASR1, RPAS(RO)	; CLEAR THE ATTENTION BIT FOR PORT B
7832	051412	113760	001224	000010	MOV B	PORTA, RPCS2(RO)	; SELECT PORT A
7833	051420	013737	001224	001234	MOV	PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7834	051426	012760	177777	000014	MOV	#-1, RPER1(RO)	; LOAD 1'S INTO RPER1 THROUGH PORT A
7835	051434	012760	177777	000040	MOV	#-1, RPER2(RO)	; LOAD 1'S INTO RPER2 THROUGH PORT A
7836	051442	012760	177777	000042	MOV	#-1, RPER3(RO)	; LOAD 1'S INTO RPER3 THROUGH PORT A
7837	051450	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT B
7838	051456	013737	001226	001234	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7839	051464	004737	052206		JSR	PC, TST42B	; CHECK THE REGISTERS THROUGH PORT B
7840							
7841							; RELEASE THE DRIVE FROM PORT B
7842							
7843	051470	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT B
7844	051476	013737	001226	001234	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7845	051504	012760	000013	000000	MOV	#13, RPCS1(RO)	; ISSUE RELEASE THROUGH PORT B
7846							
7847							; VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
7848							
7849	051512	005037	001250		CLR	RELERR	; CLEAR 'RELEASE ERROR' INDICATOR
7850	051516	012737	111700	001124	MOV	#ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT	; COMPARISON CONSTANT
7851	051524	012737	000012	001122	MOV	#RPDS1,\$BDADR	; REGISTER ADDRESS INCREMENT
7852	051532	060037	001122		ADD	RO,\$BDADR	; REGISTER BASE ADDRESS FOR TYPEOUT
7853	051536	113760	001224	000010	MOV B	PORTA, RPCS2(RO)	; SELECT PORT A
7854	051544	013737	001224	001234	MOV	PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7855	051552	016037	000012	001164	MOV	RPDS1(RO), \$TMP0	; READ STATUS REGISTER FROM PORT A
7856	051560	113760	001226	000010	MOV B	PORTB, RPCS2(RO)	; SELECT PORT B
7857	051566	013737	001226	001234	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7858	051574	016037	000012	001126	MOV	RPDS1(RO), \$BDAT	; DRIVE STATUS FROM PORT B
7859	051602	001404			BEG	66\$; BR IF STATUS FROM PORT B ZERO
7860	051604	005737	001164		TST	\$TMP0	; IS STATUS FROM PORT A ZERO ?

N11

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 143
 DZRJEA.CMB 02-NOV-76 18:40 T42 TEST PORT INTERACTION FROM PORT 'B'

7861	051610	001401				BEQ	66\$;BR IF ZERO
7862	051612	104031				ERROR	31		;REPORT DRIVE IN NEUTRAL
7863	051614	013737	001164	001126	66\$:	MOV	\$TMP0,\$BDDAT		;CHECK STATUS FROM PORT A
7864	051622	013737	001224	001234		MOV	PORTA,PTNBR		;CHANGE PORT ADDRESS FOR TYPEOUT
7865	051630	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;COMPARE WITH CONSTANT
7866	051636	001401				BEQ	67\$;BR IF OK
7867	051640	104027				ERROR	27		;REPORT REGISTER ERROR
7868	051642	000240			67\$:	NOP			
7869									
7870									;RELEASE THE DRIVE FROM PORT A
7871									
7872	051644	113760	001224	000010		MOVB	PORTA,RPCS2(RO)		;SELECT PORT A
7873	051652	013737	001224	001234		MOV	PORTA,PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7874	051660	012760	000013	000000		MOV	#13,RPCS1(RO)		;ISSUE RELEASE THROUGH PORT A
7875									
7876									;VERIFY THAT THE DRIVE IS IN NEUTRAL
7877									
7878	051666	005037	001250			CLR	RELERR		;CLEAR THE 'RELEASE ERROR' INDICATOR
7879	051672	012737	000012	001122		MOV	#RPDS1,\$BDADR		;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7880	051700	060037	001122			ADD	RO,\$BDADR		;ADD THE I/O BASE ADDRESS
7881	051704	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT		;COMPARISON CONSTANT
7882	051712	113760	001224	000010		MOVB	PORTA,RPCS2(RO)		;SELECT PORT A.
7883	051720	016037	000012	001170		MOV	RPDS1(RO),\$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.
7884	051726	013737	001170	001164		MOV	\$TMP2,\$TMP0		;COPY IT INTO '\$TMP0'
7885	051734	042737	100100	001164		BIC	#ATA!VV,\$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY
7886	051742	113760	001226	000010		MOVB	PORTB,RPCS2(RO)		;SELECT PORT B.
7887	051750	016037	000012	001172		MOV	RPDS1(RO),\$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.
7888	051756	013737	001172	001166		MOV	\$TMP3,\$TMP1		;COPY IT INTO '\$TMP1'
7889	051764	042737	100100	001166		BIC	#ATA!VV,\$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY
7890	051772	023737	001164	001166		CMP	\$TMP0,\$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7891	052000	001006				BNE	68\$;BR IF NOT
7892	052002	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7893	052006	001045				BNE	70\$;BR IF NOT
7894	052010	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7895	052012	000137	052176			JMP	72\$;BYPASS THE REST OF THE CHECKS
7896	052016	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7897	052024	013737	001226	001234		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7898	052032	113760	001226	000010		MOVB	PORTB,RPCS2(RO)		;SELECT PORT B.
7899	052040	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
7900	052044	001414				BEQ	69\$;BR IF ZERO
7901	052046	013737	001224	001234		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7902	052054	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
7903	052062	113760	001224	000010		MOVB	PORTA,RPCS2(RO)		;SELECT PORT A.
7904	052070	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
7905	052074	001012				BNE	70\$;BR IF NOT
7906	052076	012737	177777	001250	69\$:	MOV	#-1,RELERR		;SET 'RELEASE ERROR' INDICATOR
7907	052104	012760	000011	000000		MOV	#11,RPCS1(RO)		;CLEAR THE DRIVE
7908	052112	012760	000013	000000		MOV	#13,RPCS1(RO)		;RELEASE THE DRIVE
7909	052120	104026				ERROR	26		;TYPE ERROR MESSAGE 26
7910	052122	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7911	052130	013737	001224	001234		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER
7912	052136	023737	001124	001170		CMP	\$GDDAT,\$TMP2		;ALL BITS OK ?
7913	052144	001401				BEQ	71\$;BR IF OK FROM PORT A.
7914	052146	104007				ERROR	7		;REPORT ERROR
7915	052150	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT		;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7916	052156	013737	001226	001234		MOV	PORTB,PTNBR		;CHANGE PORT NUMBER

7917	052164	023737	001124	001172		CMP	\$GDDAT,\$TMP3	:SEE IF READ OK FROM PORT B.
7918	052172	001401				BEQ	72\$:BR IF OK
7919	052174	104007				ERROR	7	:REPORT ERROR
7920	052176	000240			72\$:	NOP		
7921	052200	000004				SCOPE		:LOOP ?
7922	052202	000137	052540			JMP	TST43	:GO TO THE NEXT TEST
7923								
7924								:CHECK THE REGISTERS ON THE SELECTED PORT
7925								
7926	052206				TST42\$:			
7927	052206	005037	001244			CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
7928	052212	016037	000014	001126		MOV	RPER1(RO), \$BDDAT	:GET CONTENTS OF RPER1
7929	052220	012737	000014	001122		MOV	#RPER1,\$B0ADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
7930	052226	060037	001122			ADD	RO,\$B0ADR	:ADD RH11 BASE ADDRESS
7931	052232	005037	001124			CLR	\$GDDAT	:WHAT REGISTER SHOULD BE
7932	052236	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:IS THE REGISTER OK ?
7933	052244	001403				BEQ	64\$:BR IF OK
7934	052246	104006				ERROR	6	:TYPE MESSAGE 6
7935	052250	005137	001244			COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR
7936	052254	016037	000000	001126	64\$:	MOV	RPCS1(RO), \$BDDAT	:GET THE CONTENTS OF RHCS1
7937	052262	012737	000000	001122		MOV	#RPCS1,\$B0ADR	:FORM ADDRESS OF REGISTER
7938	052270	060037	001122			ADD	RO,\$B0ADR	:ADDRESS BASE
7939	052274	032737	020000	001126		BIT	#MCPE,\$BDDAT	:IS 'MCPE' SET ?
7940	052302	001404				BEQ	65\$:BR IF NOT
7941	052304	104011				ERROR	11	:REPORT THE ERROR
7942	052306	012760	040000	000000		MOV	#TRE,RPCS1(RO)	:CLEAR 'MCPE'
7943	052314	000240			65\$:	NOP		
7944	052316	005037	001244			CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
7945	052322	016037	000040	001126		MOV	RPER2(RO), \$BDDAT	:GET CONTENTS OF RPER2
7946	052330	012737	000040	001122		MOV	#RPER2,\$B0ADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
7947	052336	060037	001122			ADD	RO,\$B0ADR	:ADD RH11 BASE ADDRESS
7948	052342	005037	001124			CLR	\$GDDAT	:WHAT REGISTER SHOULD BE
7949	052346	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:IS THE REGISTER OK ?
7950	052354	001403				BEQ	66\$:BR IF OK
7951	052356	104006				ERROR	6	:TYPE MESSAGE 6
7952	052360	005137	001244			COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR
7953	052364	016037	000000	001126	66\$:	MOV	RPCS1(RO), \$BDDAT	:GET THE CONTENTS OF RHCS1
7954	052372	012737	000000	001122		MOV	#RPCS1,\$B0ADR	:FORM ADDRESS OF REGISTER
7955	052400	060037	001122			ADD	RO,\$B0ADR	:ADDRESS BASE
7956	052404	032737	020000	001126		BIT	#MCPE,\$BDDAT	:IS 'MCPE' SET ?
7957	052412	001404				BEQ	67\$:BR IF NOT
7958	052414	104011				ERROR	11	:REPORT THE ERROR
7959	052416	012760	040000	000000		MOV	#TRE,RPCS1(RO)	:CLEAR 'MCPE'
7960	052424	000240			67\$:	NOP		
7961	052426	005037	001244			CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
7962	052432	016037	000042	001126		MOV	RPER3(RO), \$BDDAT	:GET CONTENTS OF RPER3
7963	052440	012737	000042	001122		MOV	#RPER3,\$B0ADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
7964	052446	060037	001122			ADD	RO,\$B0ADR	:ADD RH11 BASE ADDRESS
7965	052452	005037	001124			CLR	\$GDDAT	:WHAT REGISTER SHOULD BE
7966	052456	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:IS THE REGISTER OK ?
7967	052464	001403				BEQ	68\$:BR IF OK
7968	052466	104006				ERROR	6	:TYPE MESSAGE 6
7969	052470	005137	001244			COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR
7970	052474	016037	000000	001126	68\$:	MOV	RPCS1(RO), \$BDDAT	:GET THE CONTENTS OF RHCS1
7971	052502	012737	000000	001122		MOV	#RPCS1,\$B0ADR	:FORM ADDRESS OF REGISTER
7972	052510	060037	001122			ADD	RO,\$B0ADR	:ADDRESS BASE

7973 052514 032737 020000 001126
7974 052522 001404
7975 052524 104011
7976 052526 012760 040000 000000
7977 052534 000240
7978 052536 000207
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994 052540
7995 052540 005737 001274
7996 052544 001406
7997 052546 100002
7998 052550 000137 002602
7999 052554 012737 177777 001274
8000 052562 112737 000043 001102
8001 052570 012737 052612 001106
8002 052576 012737 052612 001110
8003 052604 012737 000031 001176
8004 052612 012706 001100
8005
8006
8007
8008 052616 113760 001224 000010
8009 052624 005060 000012
8010 052630 012760 000011 000000
8011 052636 012760 000013 000000
8012 052644 113760 001226 000010
8013 052652 005060 000012
8014 052656 012760 000011 000000
8015 052664 012760 000013 000000
8016 052672 113760 001224 000010
8017 052700 012760 177777 000014
8018 052706 005060 000014
8019 052712 113760 001226 000010
8020 052720 005760 000012
8021 052724 001775
8022 052726 012737 000016 001122
8023 052734 060037 001122
8024 052740 013737 001232 001124
8025 052746 013737 001232 001166
8026 052754 005137 001166
8027 052760 012737 053014 001110
8028 052766 113760 001226 000010

BIT #MCPE,\$BDDAT ;IS 'MCPE' SET ?
BEQ 69\$;BR IF NOT
ERROR 11 ;REPORT THE ERROR
MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
69\$: NOP
RTS PC ;RETURN

*TEST 43 TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
*
*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 RPDS1.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.
*

TST43:
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 #43,\$STSTM ;TEST NUMBER
MOV #TEST43,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST43,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,\$TIMES ;DO 25. ITERATIONS
TEST43: MOV #STACK,SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV #-1,RPER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT A
CLR RPER1(RO) ;CLEAR THE ERRORS
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
1\$: TST RPDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
BEQ 1\$;BR IF STILL SEIZED BY PORT A
MOV #RPAS,\$BOADR ;FORM ADDRESS OF ATTN REG IF ERROR
ADD RO,\$BOADR ;ADD THE ADDRESS BASE
MOV ASR1,\$GODAT ;GOOD DATA FOR ERROR MESSAGE
MOV ASR1,\$TMP1 ;MAKE DATA COMPARE MASK
COM \$TMP1 ;COMPLEMENT IT
MOV #25,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOVB PORTB,RPCS2(RO) ;SELECT PORT B

8029	052774	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8030	053002	013737	001226	001236		MOV	PORTB,SEIZPT	; 'SEIZED' PORT ADDRESS
8031	053010	005060	000012			CLR	RPDS1(RO)	;SEIZE THE DRIVE THROUGH PORT B
8032	053014	016037	000016	001126	2\$:	MOV	RPAS(RO), \$BDDAT	;GET THE CONTENTS OF THE ATTENTION REG
8033	053022	013737	001126	001164		MOV	\$BDDAT, \$TMP0	;PUT CONTENTS INTO WORKING LOCATION
8034	053030	043737	001166	001164		BIC	\$TMP1, \$TMP0	;CLEAR OTHER BITS
8035	053036	023737	001124	001164		CMP	\$GDDAT, \$TMP0	;SEE IF ATTN BIT FOR DRIVE SET
8036	053044	001401				BEQ	3\$;BR IF SET
8037	053046	104053				ERROR	53	;REPORT THE ERROR
8038	053050				3\$:			
8039								
8040								;RELEASE THE DRIVE FROM PORT B
8041								
8042	053050	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
8043	053056	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8044	053064	012760	000013	000000		MOV	#13,RPCS1(RO)	;ISSUE RELEASE THROUGH PORT B
8045								
8046								;VERIFY THAT THE DRIVE IS IN NEUTRAL
8047								
8048	053072	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
8049	053076	012737	000012	001122		MOV	#RPDS1, \$BDDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8050	053104	060037	001122			ADD	RO, \$BDDADR	;ADD THE I/O BASE ADDRESS
8051	053110	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;COMPARISON CONSTANT
8052	053116	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
8053	053124	016037	000012	001170		MOV	RPDS1(RO), \$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
8054	053132	013737	001170	001164		MOV	\$TMP2, \$TMP0	;COPY IT INTO '\$TMP0'
8055	053140	042737	100100	001164		BIC	#ATA!VV, \$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
8056	053146	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
8057	053154	016037	000012	001172		MOV	RPDS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
8058	053162	013737	001172	001166		MOV	\$TMP3, \$TMP1	;COPY IT INTO '\$TMP1'
8059	053170	042737	100100	001166		BIC	#ATA!VV, \$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
8060	053176	023737	001164	001166		CMP	\$TMP0, \$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8061	053204	001006				BNE	64\$;BR IF NOT
8062	053206	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8063	053212	001045				BNE	66\$;BR IF NO
8064	053214	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8065	053216	000137	053416			JMP	68\$;BYPASS THE REST OF THE CHECKS
8066	053222	013737	001170	001126	64\$:	MOV	\$TMP2, \$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8067	053230	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8068	053236	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
8069	053244	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
8070	053250	001414				BEQ	65\$;BR IF ZERO
8071	053252	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8072	053260	013737	001172	001126		MOV	\$TMP3, \$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
8073	053266	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
8074	053274	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
8075	053300	001012				BNE	66\$;BR IF NOT
8076	053302	012737	177777	001250	65\$:	MOV	#-1, RELERR	;SET 'RELEASE ERROR' INDICATOR
8077	053310	012760	000011	000000		MOV	#11, RPCS1(RO)	;CLEAR THE DRIVE
8078	053316	012760	000013	000000		MOV	#13, RPCS1(RO)	;RELEASE THE DRIVE
8079	053324	104026				ERROR	26	;TYPE ERROR MESSAGE 26
8080	053326	013737	001170	001126	66\$:	MOV	\$TMP2, \$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8081	053334	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
8082	053342	042737	100000	001170		BIC	#ATA, \$TMP2	;DON'T CHECK THE ATTN BIT
8083	053350	023737	001124	001170		CMP	\$GDDAT, \$TMP2	;ALL BITS OK ?
8084	053356	001401				BEQ	67\$;BR IF OK FROM PORT A.

E12

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 147
DZRJE.A.CMB 02-NOV-76 18:40 T43 TEST PORT 'A' ALTERNATE ATTENTION BIT PATH

```
8085 053360 104007          ERROR 7          ;REPORT ERROR
8086 053362 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PCPT B.
8087 053370 013737 001226 001234      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8088 053376 042737 100000 001172      BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
8089 053404 023737 001124 001172      CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8090 053412 001401          BEQ 68$          ;BR IF OK
8091 053414 104007          ERROR 7          ;REPORT ERROR
8092 053416 000240          NOP
8093 053420 000004          SCOPE          ;LOOP ?
```

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

*TEST 44 TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
*
*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 RPDS1.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.

```
8108 053422          TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
8109 053422 005737 001274      BEQ 2$      ;BR IF NOT
8110 053426 001406          BPL 1$      ;BR IF JUST ENTERED TEST
8111 053430 100002          JMP EXEC    ;RETURN & GET NEXT TEST NUMBER
8112 053432 000137 002602      MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
8113 053436 012737 177777 001274 1$: MOV #44,$STNM ;TEST NUMBER
8114 053444 112737 000044 001102 2$: MOV #TEST44,$LPADR ;LOAD LOOP ON TEST ADDRESS
8115 053452 012737 053474 001106      MOV #TEST44,$LPERR ;LOAD LOOP ON ERROR ADDRESS
8116 053460 012737 053474 001110      MOV #25,$TIMES ;DO 25. ITERATIONS
8117 053466 012737 000031 001176      MOV #STACK,SP ;LOAD THE STACK POINTER
8118 053474 012706 001100      TEST44: MOV
```

```
8119
8120
8121
8122 053500 113760 001224 000010      MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
8123 053506 005060 000012          CLR RPDS1(RO) ;SEIZE THE DRIVE
8124 053512 012760 000011 000000      MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
8125 053520 012760 000013 000000      MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8126 053526 113760 001226 000010      MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
8127 053534 005060 000012          CLR RPDS1(RJ) ;SEIZE THE DRIVE THROUGH PORT 'B'
8128 053540 012760 000011 000000      MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
8129 053546 012760 000013 000000      MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8130 053554 113760 001226 000010      MOVB PORTB,RPCS2(RO) ;SELECT PORT B
8131 053562 012760 177777 000014      MOV #-1,RPER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT B
8132 053570 005060 000014          CLR RPER1(RO) ;CLEAR THE ERRORS
8133 053574 113760 001224 000010      MOVB PORTA,RPCS2(RO) ;SELECT PORT A
8134 053602 005760 000012          TST RPDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
8135 053606 001775          BEQ 1$      ;BR IF STILL SEIZED BY PORT B
8136 053610 012737 000016 001122      MOV #RPAS,$BDADR ;FORM ADDRESS OF ATTN REG IF ERROR
8137 053616 060037 001122          ADD RO,$BDADR ;ADD THE ADDRESS BASE
8138 053622 013737 001232 001124      MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
8139 053630 013737 001232 001166      MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
8140 053636 005137 001166          COM $TMP1 ;COMPLEMENT IT
```

F12

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 148
 DZRJEA.CMB 02-NOV-76 18:40 T44 TEST PORT 'B' ALTERNATE ATTENTION BIT PATH

8141	053642	012737	053676	001110		MOV #25,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
8142	053650	113760	001224	000010		MOV PORTA,RPCS2(RO) ;SELECT PORT A
8143	053656	013737	001224	001234		MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8144	053664	013737	001224	001236		MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
8145	053672	005060	000012			CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT A
8146	053676	016037	000016	001126	25:	MOV RPAS(RO),\$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
8147	053704	013737	001126	001164		MOV \$BDDAT,\$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
8148	053712	043737	001166	001164		BIC \$TMP1,\$TMP0 ;CLEAR OTHER BITS
8149	053720	023737	001124	001164		CMP \$GDDAT,\$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
8150	053726	001401				BEQ 35 ;BR IF SET
8151	053730	104053				ERROR 53 ;REPORT THE ERROR
8152	053732				35:	
8153						
8154						;RELEASE THE DRIVE FROM PORT A
8155						
8156	053732	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A
8157	053740	013737	001224	001234		MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8158	053746	012760	000013	000000		MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
8159						
8160						;VERIFY THAT THE DRIVE IS IN NEUTRAL
8161						
8162	053754	005037	001250			CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8163	053760	012737	000012	001122		MOV #RPDS1,\$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8164	053766	060037	001122			ADD RO,\$BDAOR ;ADD THE I/O BASE ADDRESS
8165	053772	012737	011700	001124		MOV #MOL:PGM:DPR:DRY:VV,\$GDDAT ;COMPARISON CONSTANT
8166	054000	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8167	054006	016037	000012	001170		MOV RPDS1(RO),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8168	054014	013737	001170	001164		MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
8169	054022	042737	100100	001164		BIC #ATA:VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8170	054030	113760	001226	000010		MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8171	054036	016037	000012	001172		MOV RPDS1(RO),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8172	054044	013737	001172	001166		MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
8173	054052	042737	100100	001166		BIC #ATA:VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8174	054060	023737	001164	001166		CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8175	054066	001006				BNE 645 ;BR IF NOT
8176	054070	005737	001164			TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8177	054074	001045				BNE 665 ;BR IF NOT
8178	054076	104046				ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8179	054100	000137	054300			JMP 685 ;BYPASS THE REST OF THE CHECKS
8180	054104	013737	001170	001126	645:	MOV \$TMP2,\$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8181	054112	013737	001226	001234		MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8182	054120	113760	001226	000010		MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8183	054126	005737	001164			TST \$TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8184	054132	001414				BEQ 655 ;BR IF ZERO
8185	054134	013737	001224	001234		MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8186	054142	013737	001172	001126		MOV \$TMP3,\$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8187	054150	113760	001224	000010		MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8188	054156	005737	001166			TST \$TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8189	054162	001012				BNE 665 ;BR IF NOT
8190	054164	012737	177777	001250	655:	MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8191	054172	012760	000011	000000		MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
8192	054200	012760	000013	000000		MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8193	054206	104026				ERROR 26 ;TYPE ERROR MESSAGE 26
8194	054210	013737	001170	001126	665:	MOV \$TMP2,\$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8195	054216	013737	001224	001234		MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8196	054224	042737	100000	001170		BIC #ATA,\$TMP2 ;DON'T CHECK THE ATTN BIT

G12

```

8197 054232 023737 001124 001170      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
8198 054240 001401                BEQ      67$                ;BR IF OK FROM PORT A.
8199 054242 104007                ERROR    7                  ;REPORT ERROR
8200 054244 013737 001172 001126 67$:  MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8201 054252 013737 001226 001234      MOV      PORTB,PTNBR       ;CHANGE PORT NUMBER
8202 054260 042737 100000 001172      BIC      #ATA,$TMP3        ;DON'T CHECK THE ATTN BIT
8203 054266 023737 001124 001172      CMP      $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
8204 054274 001401                BEQ      68$                ;BR IF OK
8205 054276 104007                ERROR    7                  ;REPORT ERROR
8206 054300 000240                NOP
8207 054302 000004                SCOPE
8208 054304 000137 056224                JMP      $EOP                ;LOOP ?
                                   ;GO TO END OF TEST

```

;;*****

.SBTTL *** SPECIAL TESTS FOR THE M7775 ('DP') BOARD ***

;;*****

;;*****

```

*TEST 45      TEST NO TIMEOUT THROUGH PORT 'A'
*
*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT 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 RPDS1.
*
* B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 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 NOT
*BEEN RELEASED.
*
* E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
*RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

```

;;*****

```

8238 054310                TST      KYBCTL              ;PERFORMING ONLY SINGLE TESTS ?
8239 054310 005737 001274      BEQ      25                  ;BR IF NOT
8240 054314 001406                BPL      15                  ;BR IF JUST ENTERED TEST
8241 054316 100002                JMP      EXEC                ;RETURN & GET NEXT TEST NUMBER
8242 054320 000137 002602      MOV      #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
8243 054324 012737 177777 001274 1$:  MOV      #45,$STSTNM         ;TEST NUMBER
8244 054332 112737 000045 001102 2$:  MOV      #TEST45,$LPADR      ;LOAD LOOP ON TEST ADDRESS
8245 054340 012737 054362 001106      MOV      #TEST45,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
8246 054346 012737 054362 001110      MOV      #4,$TIMES          ;DO 4 ITERATIONS
8247 054354 012737 000004 001176      MOV      #STACK,$SP         ;LOAD THE STACK POINTER
8248 054362 012706 001100      TEST45: MOV
8249
8250                ;CLEAR ATTENTION BITS FOR BOTH PORTS
8251
8252 054366 113760 001224 000010      MOV      PORTA,RPCS2(RO)    ;SELECT PORT #A

```

H12

MO-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 150
DZRJEA.CMB 02-NOV-76 18:40 T45 TEST NO TIMEOUT THROUGH PORT 'A'

```
8253 054374 005060 000012          CLR      RPDS1(RO)      ;SEIZE THE DRIVE
8254 054400 012760 000011 000000    MOV      #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
8255 054406 012760 000013 000000    MOV      #13,RPDS1(RO) ;RELEASE THE DRIVE
8256 054414 113760 001226 000010    MOVB     PORTB,RPDS2(RO);SELECT PORT #B
8257 054422 005060 000012          CLR      RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
8258 054426 012760 000011 000000    MOV      #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
8259 054434 012760 000013 000000    MOV      #13,RPDS1(RO) ;RELEASE THE DRIVE
8260
8261                                     ;:*****
8262
8263                                     ;SEIZE THE DRIVE THROUGH PORT B
8264
8265 054442 113760 001226 000010    MOVB     PORTB,RPDS2(RO);SELECT PORT B
8266 054450 013737 001226 001236    MOV      PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
8267 054456 005060 000012          CLR      RPDS1(RO)      ;WRITE RPDS1
8268 054462 013737 001224 001240    MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
8269 054470 113760 001224 000010    MOVB     PORTA,RPDS2(RO);SELECT PORT A
8270 054476 013737 001224 001234    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8271
8272                                     ;:*****
8273                                     ;SET REQUEST THROUGH PORT A
8274
8275 054504 005060 000012          CLR      RPDS1(RO)      ;SET REQUEST FOR PORT A
8276 054510 113760 001226 000010    MOVB     PORTB,RPDS2(RO);SELECT PORT B
8277 054516 013737 001226 001234    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8278
8279                                     ;:*****
8280                                     ;RELEASE THE DRIVE THROUGH PORT B
8281
8282 054524 012760 000013 000000    MOV      #13,RPDS1(RO) ;RELEASE DRIVE THROUGH PORT B
8283
8284                                     ;:*****
8285                                     ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8286
8287 054532 013737 001260 001254    MOV      TIMEAP,WATCH  ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8288
8289                                     ;:*****
8290                                     ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
8291
8292 054540 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8293 054544 016037 000012 001126    MOV      RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
8294 054552 012737 000012 001122    MOV      #RPDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8295 054550 060037 001122          ADD      RD,SBADR       ;ADD RH11 BASE ADDRESS
8296 054564 005037 001124          CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
8297 054570 023737 001124 001126    CMP      $GDDAT,SBDDAT  ;IS THE REGISTER OK ?
8298 054576 001403          BEQ      66$           ;BR IF OK
8299 054600 104031          ERROR    31           ;TYPE MESSAGE 31
8300 054602 005137 001244          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
8301 054606 000240          NOP
8302 054610 005737 001244          TST     CKERR          ;REGISTER OK ?
8303 054614 001402          BEQ      .+6          ;BR IF OK
8304 054616 000137 055254          JMP     IS             ;BYPASS REST OF TEST IF NOT
8305 054622 005737 001254          TST     WATCH         ;WATCH EQUAL ZERO ?
8306 054626 001375          BNE     .-4           ;BR IF NOT
8307
8308                                     ;:*****
```


;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT

```

8309
8310
8311 054630 013737 001224 001234      MOV      PORTA,PTNBR      ;PORT NUMBER FOR TYPEOUT
8312 054636 005037 001244          CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8313 054642 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
8314 054650 012737 000012 001122      MOV      #RPDS1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8315 054656 060037 001122          ADD      RO, $BDAADR     ;ADD RHI1 BASE ADDRESS
8316 054662 005037 001124          CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
8317 054666 023737 001124 001126      CMP      $GDDAT, $BDDAT  ;IS THE REGISTER OK ?
8318 054674 001403          BEQ     68$            ;BR IF OK
8319 054676 104035          ERROR   35            ;TYPE MESSAGE 35
8320 054700 005137 001244          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8321 054704 000240          NOP
8322 054706 005737 001244          TST     CKERR          ;REGISTER OK ?
8323 054712 001402          BEQ     +6            ;BR IF OK
8324 054714 000137 055254          JMP     1$            ;BYPASS REST OF TEST IF NOT
8325
8326
8327
8328
8329
8330 054720 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
8331 054726 013737 001224 001234      MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8332 054734 012760 000013 000000      MOV     #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT A
8333
8334
8335
8336 054742 005037 001250          CLR     RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
8337 054746 012737 000012 001122      MOV     #RPDS1, $BDAADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8338 054754 060037 001122          ADD     RO, $BDAADR    ;ADD THE I/O BASE ADDRESS
8339 054760 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
8340 054766 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
8341 054774 016037 000012 001170      MOV     RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8342 055002 013737 001170 001164      MOV     $TMP2, $TMP0    ;COPY IT INTO 'TMP0'
8343 055010 042737 100100 001164      BIC     #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8344 055016 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
8345 055024 016037 000012 001172      MOV     RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8346 055032 013737 001172 001166      MOV     $TMP3, $TMP1    ;COPY IT INTO 'TMP1'
8347 055040 042737 100100 001166      BIC     #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8348 055046 023737 001164 001166      CMP     $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8349 055054 001006          BNE     70$           ;BR IF NOT
8350 055056 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8351 055062 001045          BNE     72$           ;BR IF NOT
8352 055064 104046          ERROR   46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8353 055066 000137 055252          JMP     74$           ;BYPASS THE REST OF THE CHECKS
8354 055072 013737 001170 001126 70$:  MOV     $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8355 055100 013737 001226 001234      MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8356 055106 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
8357 055114 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
8358 055120 001414          BEQ     71$           ;BR IF ZERO
8359 055122 013737 001224 001234      MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8360 055130 013737 001172 001126      MOV     $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
8361 055136 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
8362 055144 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
8363 055150 001012          BNE     72$           ;BR IF NOT
8364 055152 012737 177777 001250 71$:  MOV     #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR

```

;*****

:RELEASE THE DRIVE FROM PORT A

J12

```

8365 055160 012760 000011 000000      MOV      #11,RPCS1(R0)      ;CLEAR THE DRIVE
8366 055166 012760 000013 000000      MOV      #13,RPCS1(R0)      ;RELEASE THE DRIVE
8367 055174 104026                ERROR      26                ;TYPE ERROR MESSAGE 26
8368 055176 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8369 055204 013737 001224 001234      MOV      PORTA,PTNBR        ;CHANGE PORT NUMBER
8370 055212 023737 001124 001170      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
8371 055220 001401                BEQ      73$                ;BR IF OK FROM PORT A.
8372 055222 104007                ERROR      7                ;REPORT ERROR
8373 055224 013737 001172 001126 73$:  MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8374 055232 013737 001226 001234      MOV      PORTB,PTNBR        ;CHANGE PORT NUMBER
8375 055240 023737 001124 001172      CMP      $GDDAT,$TMP3      ;SEL IF READ OK FROM PORT B.
8376 055246 001401                BEQ      74$                ;BR IF OK
8377 055250 104007                ERROR      7                ;REPORT ERROR
8378 055252 000240                74$:  NOP
8379
8380 055254 000004                1$:  SCOPE                    ;LOOP ?
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420

```

```

*****
*TEST 46      TEST NO TIMEOUT THROUGH PORT 'B'
*
*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT 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 RPDS1.
*
* B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 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 NOT
*BEEN RELEASED.
*
* E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
*RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****

```

```

TST46:
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     #46,$STNM         ;TEST NUMBER
      MOV      #TEST46,$LPADR   ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST46,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
      MOV      #4,$TIMES        ;DO 4 ITERATIONS
TEST46: MOV      #STACK,SP      ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB     PORTA,RPCS2(R0)       ;SELECT PORT #A
CLR      RPDS1(R0)            ;SEIZE THE DRIVE
MOV      #11,RPCS1(R0)        ;ISSUE DRIVE CLEAR
MOV      #13,RPCS1(R0)        ;RELEASE THE DRIVE

```

K12

8421 055362 113760 001226 000010
8422 055370 005060 000012
8423 055374 012760 000011 000000
8424 055402 012760 000013 000000
8425
8426
8427
8428
8429
8430 055410 113760 001224 000010
8431 055416 013737 001224 001236
8432 055424 005060 000012
8433 055430 013737 001226 001240
8434 055436 113760 001226 000010
8435 055444 013737 001226 001234
8436
8437
8438
8439
8440 055452 005060 000012
8441 055456 113760 001224 000010
8442 055464 013737 001224 001234
8443
8444
8445
8446
8447 055472 012760 000013 000000
8448
8449
8450
8451
8452 055500 013737 001266 001254
8453
8454
8455
8456
8457 055506 005037 001244
8458 055512 016037 000012 001126
8459 055520 012737 000012 001122
8460 055526 060037 001122
8461 055532 005037 001124
8462 055536 023737 001124 001126
8463 055544 001403
8464 055546 104031
8465 055550 005137 001244
8466 055554 000240
8467 055556 005737 001244
8468 055562 001402
8469 055564 000137 056222
8470 055570 005737 001254
8471 055574 001375
8472
8473
8474
8475
8476 055576 013737 001226 001234

MOV B PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
;*****
;SEIZE THE DRIVE THROUGH PORT A
MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;*****
;SET REQUEST THROUGH PORT B
CLR RPDS1(RO) ;SET REQUEST FOR PORT B
MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;*****
;RELEASE THE DRIVE THROUGH PORT A
MOV #13,RPCS1(RO) ;RELEASE DRIVE THROUGH PORT A
;*****
;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
MOV TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
;*****
;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), \$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,\$BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,\$BDAADR ;ADD RH11 BASE ADDRESS
CLR \$GDDAT ;WHAT REGISTER SHOULD BE
CMP \$GDDAT,\$BDDAT ;IS THE REGISTER OK ?
BEQ 66\$;BR IF OK
ERROR 31 ;TYPE MESSAGE 31
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
66\$: NOP
TST CKERR ;REGISTER OK ?
BEQ .+6 ;BR IF OK
JMP 1\$;BYPASS REST OF TEST IF NOT
TST WATCH ;WATCH EQUAL ZERO ?
BNE .-4 ;BR IF NOT
;*****
;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
MOV PORTB,PTNBR ;PORT NUMBER FOR TYPEOUT

L12

```

8477 055604 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
8478 055610 016037 000012 001126    MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
8479 055616 012737 000012 001122    MOV      #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8480 055624 060037 001122          ADD      RO, $BDADR      ;ADD RH11 BASE ADDRESS
8481 055630 005037 001124          CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
8482 055634 023737 001124 001126    CMP      $GDDAT, $BDDAT  ;IS THE REGISTER OK ?
8483 055642 001403          BEQ      68$           ;BR IF OK
8484 055644 104035          ERROR    35           ;TYPE MESSAGE 35
8485 055646 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
8486 055652 000240          NOP
8487 055654 005737 001244          TST      CKERR          ;REGISTER OK ?
8488 055660 001402          BEQ      +6           ;BR IF OK
8489 055662 000137 056222          JMP      1$           ;BYPASS REST OF TEST IF NOT
8490
8491
8492
8493
8494
8495 055666 113760 001226 000010    MOV      PORTB, RPCS2(RO) ;SELECT PORT B
8496 055674 013737 001226 001234    MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8497 055702 012760 000013 0000C0    MOV      #13, RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
8498
8499
8500
8501 055710 005037 001250          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
8502 055714 012737 000012 001122    MOV      #RPDS1, $BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8503 055722 060037 001122          ADD      RO, $BDADR      ;ADD THE I/O BASE ADDRESS
8504 055726 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
8505 055734 113760 001224 000010    MOV      PORTA, RPCS2(RO) ;SELECT PORT A.
8506 055742 016037 000012 001170    MOV      RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8507 055750 013737 001170 001164    MOV      $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
8508 055756 042737 100100 001164    BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8509 055764 113760 001226 000010    MOV      PORTB, RPCS2(RO) ;SELECT PORT B.
8510 055772 016037 000012 001172    MOV      RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8511 056000 013737 001172 001166    MOV      $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
8512 056006 042737 100100 001166    BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8513 056014 023737 001164 001166    CMP      $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8514 056022 001006          BNE      70$           ;BR IF NOT
8515 056024 005737 001164          TST      $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8516 056030 001045          BNE      72$           ;BR IF NOT
8517 056032 104046          ERROR    46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8518 056034 000137 056220          JMP      74$           ;BYPASS THE REST OF THE CHECKS
8519 056040 013737 001170 001126 70$:  MOV      $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8520 056046 013737 001226 001234    MOV      PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8521 056054 113760 001226 000010    MOV      PORTB, RPCS2(RO) ;SELECT PORT B.
8522 056062 005737 001164          TST      $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
8523 056066 001414          BEQ      71$           ;BR IF ZERO
8524 056070 013737 001224 001234    MOV      PORTA, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8525 056076 013737 001172 001126    MOV      $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
8526 056104 113760 001224 000010    MOV      PORTA, RPCS2(RO) ;SELECT PORT A.
8527 056112 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
8528 056116 001012          BNE      72$           ;BR IF NOT
8529 056120 012737 177777 001250 71$:  MOV      #-1, RELERR     ;SET 'RELEASE ERROR' INDICATOR
8530 056126 012760 000011 000000    MOV      #11, RPCS1(RO)  ;CLEAR THE DRIVE
8531 056134 012760 000013 000000    MOV      #13, RPCS1(RO)  ;RELEASE THE DRIVE
8532 056142 104026          ERROR    26           ;TYPE ERROR MESSAGE 26

```

M12

```

8533 056144 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8534 056152 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8535 056160 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
8536 056166 001401 BEQ 73$ ;BR IF OK FROM PORT A.
8537 056170 104007 ERROR 7 ;REPORT ERROR
8538 056172 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8539 056200 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8540 056206 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8541 056214 001401 BEQ 74$ ;BR IF OK
8542 056216 104007 ERROR 7 ;REPORT ERROR
8543 056220 000240 74$: NOP
8544
8545 056222 000004 1$: SCOPE ;LOOP ?
8546
8547 .SBTTL END OF PASS ROUTINE
8548
8549 ;*****
8550 ;*INCREMENT THE PASS NUMBER ($PASS)
8551 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
8552 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
8553 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
8554 ;*IF THERES A MONITOR GO TO IT
8555 ;*IF THERE ISN'T JUMP TO TSTIAA
8556
8557 056224 $EOP: TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
8558 056224 005737 001274 BEQ .+6 ;BR IF NOT
8559 056230 001402 JMP EXEC ;RETURN TO KEYBOARD CONTROL
8560 056232 000137 002602 CLR $TSTNM ;ZERO THE TEST NUMBER
8561 056236 005037 001102 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
8562 056242 005037 001176 INC $PASS ;INCREMENT THE PASS NUMBER
8563 056246 005237 001100 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
8564 056252 042737 100000 001100 DEC (PC)+ ;LOOP?
8565 056260 005327 $EOPCT: .WORD 1
8566 056262 000001 BGT $DOAGN ;YES
8567 056264 003063 MOV (PC)+,2(PC)+ ;RESTORE COUNTER
8568 056266 012737 $ENDCT: .WORD 1
8569 056270 000001 TYPE ,65$ ;TYPE ASCIZ STRING
8570 056272 056262 BR ,64$ ;GET OVER THE ASCIZ
8571 056274 104401 056302 ;:65$: .ASCIZ <12><15>/END PASS #/
8572 056300 000407 ;:64$:
8573
8574 056320 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
8575 056320 013746 001100 ;:TYPE PASS NUMBER
8576 ;GO TYPE--DECIMAL ASCII WITH SIGN
8577 056324 104405 TYPDS ;TYPE ASCIZ STRING
8578 056326 104401 056334 BR ,66$ ;GET OVER THE ASCIZ
8579 056332 000421 ;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
8580 ;:66$:
8581 056376 MOV $ERTTL,-(SP) ;SAVE $ERTTL FOR TYPEOUT
8582 056376 013746 001112 ;TOTAL NUMBER OF ERRORS
8583 ;GO TYPE--DECIMAL ASCII WITH SIGN
8584 056402 104405 TYPDS ;TYPE CARRIAGE RETURN, LINE FEED
8585 056404 104401 001207 TYPE ,$CRLF
8586 056410 005037 001112 CLR $ERTTL ;CLEAR ERROR TOTAL
8587 056414 013700 000042 $GET42: MOV ,2*42,RO ;GET MONITOR ADDRESS
8588 056420 001405 BEQ $DOAGN ;BRANCH IF NO MONITOR

```

```

8589 056422 000005
8590 056424 004710
8591 056426 000240
8592 056430 000240
8593 056432 000240
8594 056434
8595 056434 000137
8596 056436 003064
8597 056440 377 377 000
8598 056444
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610 056444 012737 056514 000004 CKCLK: MOV #CKCLK1, @#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
8611 056452 005037 000006 CLR @#ERRVEC+2 ;NEW PSW
8612 056456 005777 122530 TST @SLKCSR ;CHECK FOR KW11-P
8613 056462 013701 001216 MOV $LPVEC, R1 ;KW11-P VECTOR ADDRESS
8614 056466 012721 056576 MOV #CLOCK, (R1)+ ;SET UP KW11-P VECTOR
8615 056472 012711 000300 MOV #300, (R1) ;PSW - PRI 6
8616 056476 012777 177777 122510 MOV #-1, @SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
8617 056504 012777 000135 122500 MOV #135, @SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
8618 056512 000425 BR CKCLK3
8619 056514 062706 000004 CKCLK1: ADD #4, SP ;RESTORE THE STACK POINTER
8620 056520 012737 056556 000004 MOV #CKCLK2, @#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
8621 056526 005777 122466 TST @SLKS ;LOOK FOR KW11-L
8622 056532 013701 001222 MOV $LLVEC, R1 ;KW11-L VECTOR ADDRESS
8623 056536 012721 056576 MOV #CLOCK, (R1)+ ;SET UP KW11-L VECTOR
8624 056542 012711 000300 MOV #300, (R1) ;PSW - PRI 6
8625 056546 012777 000100 122444 MOV #100, @SLKS ;SET KW11-L INTERRUPT
8626 056554 000404 BR CKCLK3
8627 056556 062706 000004 CKCLK2: ADD #4, SP ;RESTORE THE STACK POINTER
8628 056562 062716 000002 ADD #2, (SP) ;INCREMENT RETURN, NO CLOCK
8629 056566 012737 000006 000004 CKCLK3: MOV #6, @#ERRVEC ;RESTORE THE ERROR VECTOR
8630 056574 000207 RTS PC
8631
8632 ;ROUTINE TO COUNT CLOCK TICKS
8633
8634 056576 062737 000021 001252 CLOCK: ADD #17, TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
8635 056604 005737 001254 TST WATCH ;IS WATCH ALREADY ZERO ?
8636 056610 001406 BEQ 1$ ;BR IF IT IS
8637 056612 162737 000021 001254 SUB #17, WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
8638 056620 100002 BPL 1$ ;BR IF NOT MINUS
8639 056622 005037 001254 CLR WATCH ;CLEAR WATCH DOG COUNTER
8640 056626 000002 1$: RTI ;RETURN
8641
8642 ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
8643
8644 056630 162706 000004 TOLER: SUB #4, SP ;SETUP STACK

```

```

8645 056634 016616 000004      MOV      4(SP), (SP)      ;SAVE STACK
8646 056640 013546             MOV      2(R5)+, -(SP)   ;GET TIME VALUE
8647 056642 011666 000004      MOV      (SP), 4(SP)    ;MOVE TIME VALUE
8648 056646 011666 000006      MOV      (SP), 6(SP)    ;MOVE VALUE AGAIN
8649 056652 006216             ASR      (SP)           ;DIVIDE BY 2
8650 056654 005216             ASR      (SP)           ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
8651 056656 061666 000004      ADD      (SP), 4(SP)    ;CALCULATE UPPER LIMIT FOR TIMEOUT
8652 056662 162666 000004      SUB      (SP)+, 4(SP)   ;CALCULATE LOWER LIMIT FOR TIMEOUT
8653 056666 000205             RTS                    ;RETURN WITH TOLERANCES ON THE STACK
8654
8655 ;*****
8656
8657 .SBTTL 'SYSMAC' UTILITY ROUTINES
8658
8659 ;*****
8660
8661 .SBTTL SCOPE HANDLER ROUTINE
8662
8663 ;*****
8664 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8665 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG. (DISPLAY<7:0>)
8666 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8667 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8668 ;*SW14=1      LOOP ON TEST
8669 ;*SW11=1      INHIBIT ITERATIONS
8670 ;*SW09=1      LOOP ON ERROR
8671 ;*CALL
8672 ;*          SCOPE          ;;SCOPE=IOT
8673
8674 $SCOPE:
8675 056670 104407             CKSWR
8676 056672 032777 040000 122240 15:  BIT      #BIT14, $SWR    ;;TEST FOR CHANGE IN SOFT-SWR
8677 056700 001101             BNE      $OVER          ;;LOOP ON PRESENT TEST?
8678 ;*****START OF CODE FOR THE XOR TESTER*****      ;;YES IF SW14=1
8679 056702 000416             $XTSTR: BR      6$      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
8680 ;*****END OF CODE FOR THE XOR TESTER*****      ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
8681 056704 013746 000004      MOV      2#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
8682 056710 012737 056730 000004      MOV      #5$, 2#ERRVEC  ;;SET FOR TIMEOUT
8683 056716 005737 177060             TST      2#177060      ;;TIME OUT ON XOR?
8684 056722 012637 000004      MOV      (SP)+, 2#ERRVEC ;;RESTORE THE ERROR VECTOR
8685 056726 000453             BR      $$VLAD         ;;GO TO THE NEXT TEST
8686 056730 022626             5$:  CMP      (SP)+, (SP)+  ;;CLEAR THE STACK AFTER A TIME OUT
8687 056732 012637 000004      MOV      (SP)+, 2#ERRVEC ;;RESTORE THE ERROR VECTOR
8688 056736 000413             BR      7$           ;;LOOP ON THE PRESENT TEST
8689 056740             6$: ;*****END OF CODE FOR THE XOR TESTER*****
8690 056740 105737 001103             2$:  TSTB     $ERFLG      ;;HAS AN ERROR OCCURRED?
8691 056744 001421             BEQ      3$           ;;BR IF NO
8692 056746 123737 001115 001103      CMPB    $ERMAX, $ERFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
8693 056754 101015             BHI     3$           ;;BR IF NO
8694 056756 032777 001000 122154      BIT     #BIT09, $SWR    ;;LOOP ON ERROR?
8695 056764 001404             BEQ     4$           ;;BR IF NO
8696 056766 013737 001110 001106      7$:  MOV      $LPERR, $LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
8697 056774 000443             BR      $OVER
8698 056776 105037 001103             4$:  CLRB     $ERFLG      ;;ZERO THE ERROR FLAG
8699 057002 005037 001176             CLR     $TIMES        ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
8700 057006 000415             BR      1$           ;;ESCAPE TO THE NEXT TEST

```

```

8701 057010 032777 004000 122122 3$: BIT #BIT11, @SWR ;;INHIBIT ITERATIONS?
8702 057016 001011 BNE 1$ BR IF YES
8703 057020 005737 001100 TST $PASS ;;IF FIRST PASS OF PROGRAM
8704 057024 001406 BEQ 1$ INHIBIT ITERATIONS
8705 057026 005237 001104 INC $ICNT INCREMENT ITERATION COUNT
8706 057032 023737 001176 001104 CMP $TIMES, $ICNT CHECK THE NUMBER OF ITERATIONS MADE
8707 057040 002021 BGE $OVER BR IF MORE ITERATION REQUIRED
8708 057042 012737 000001 001104 1$: MOV #1, $ICNT REINITIALIZE THE ITERATION COUNTER.
8709 057050 013737 057120 001176 MOV $MXCNT, $TIMES SET NUMBER OF ITERATIONS TO DO
8710 057056 105237 001102 $SVLAD: INCB $STNM COUNT TEST NUMBERS
8711 057062 011637 001106 MOV (SP), $LPADR SAVE SCOPE LOOP ADDRESS
8712 057066 011637 001110 MCV (SP), $LPERR SAVE ERROR LOOP ADDRESS
8713 057072 005037 001200 CLR $ESCAPE CLEAR THE ESCAPE FROM ERROR ADDRESS
8714 057076 112737 000001 001115 MOVB #1, $ERMAX ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8715 057104 013777 001102 122030 $OVER: MOV $STNM, @DISPLAY DISPLAY TEST NUMBER
8716 057112 013716 001106 MOV $LPADR, (SP) FUDGE RETURN ADDRESS
8717 057116 000002 RTI FIXES PS
8718 057120 000004 $MXCNT: 4 MAX. NUMBER OF ITERATIONS
8719 .SBTTL ERROR HANDLER ROUTINE

```

```

8720
8721 *****
8722 *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
8723 *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8724 *AND GO TO $ERRTYP ON ERROR
8725 *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8726 *$SW15=1 HALT ON ERROR
8727 *$SW13=1 INHIBIT ERROR TYPEOUTS
8728 *$SW10=1 BELL ON ERROR
8729 *CALL
8730 * ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
8731

```

```

8732 057122 $ERROR: CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
8733 057122 104407 MOVB $STNM, TSTNUM
8734 057124 113737 001102 001242 7$: INCB $ERFLG ;;SET THE ERROR FLAG
8735 057132 105237 001103 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
8736 057136 001775 MOV $STNM, @DISPLAY DISPLAY TEST NUMBER AND ERROR FLAG
8737 057140 013777 001102 121774 BIT #BIT10, @SWR BELL ON ERROR?
8738 057146 032777 002000 121764 BEQ 1$ NO - SKIP
8739 057154 001402 TYPE $BELL RING BELL
8740 057156 104401 001202 INC $ERTTL COUNT THE NUMBER OF ERRORS
8741 057162 005237 001112 1$: MOV (SP), $ERRPC GET ADDRESS OF ERROR INSTRUCTION
8742 057166 011637 001116 SUB #2, $ERRPC
8743 057172 162737 000002 001116 MOVB @ERRPC, $ITEMB STRIP AND SAVE THE ERROR ITEM CODE
8744 057200 117737 121712 001114 BIT #BIT13, @SWR SKIP TYPEOUT IF SET
8745 057206 032777 020000 121724 BNE 20$ SKIP TYPEOUTS
8746 057214 001004 JSR PC, $ERRTYP GO TO USER ERROR ROUTINE
8747 057216 004737 057254 TYPE , $CRLF
8748 057222 104401 001207
8749 057226 20$:
8750 057226 005777 121706 2$: TST @SWR ;;HALT ON ERROR
8751 057232 100002 BPL 3$ ;;SKIP IF CONTINUE
8752 057234 000000 HALT ;;HALT ON ERROR!
8753 057236 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
8754 057240 3$:
8755 057240 022737 056424 000042 CMP #SENDAD, @#42 ;;ACT-11 AUTO-ACCEPT?
8756 057246 001001 BNE 6$ ;;BRANCH IF NO

```



```

8757 057250 000000
8758 057252
8759 057252 000002
8760
8761
8762
8763
8764
8765
8766
8767 057254
8768 057254 104401 001207
8769 057260 010046
8770 057262 005000
8771 057264 153700 001114
8772 057270 001004
8773
8774 057272 013746 001116
8775
8776 057276 104402
8777 057300 000445
8778 057302 005300
8779 057304 006300
8780 057306 006300
8781 057310 006300
8782 057312 062700 001304
8783 057316 012037 057326
8784 057322 001404
8785 057324 104401
8786 057326 000000
8787 057330 104401 001207
8788 057334 012037 057344
8789 057340 001404
8790 057342 104401
8791 057344 000000
8792 057346 104401 001207
8793 057352 010146
8794 057354 012001
8795 057356 001415
8796 057360 012000
8797 057362 105720
8798 057364 001003
8799 057366 013146
8800 057370 104402
8801 057372 000402
8802 057374
8803 057374 013146
8804 057376 104405
8805 057400 005711
8806 057402 001403
8807 057404 104401 057424
8808 057410 000764
8809
8810 057412 012601
8811 057414 012600
8812 057416 104401 001207

```

```

        HALT                ;;YES
6$:      RTI                 ;;RETURN
        .SBTTL  ERROR MESSAGE TIMEOUT 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.
;*****
$ERRTYP:
        TYPE      $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
        MOV       RO, -(SP)  ;; SAVE RO
        CLR       RO         ;; PICKUP THE ITEM INDEX
        BISB     2*$ITEMB, RO
        BNE      1$         ;; IF ITEM NUMBER IS ZERO, JUST
                           ;; TYPE THE PC OF THE ERROR
        MOV       $ERRPC, -(SP)
                           ;; SAVE $ERRPC FOR TIMEOUT
                           ;; ERROR ADDRESS
        TYPDC    10$        ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR       1$         ;; GET OUT
1$:      DEC      RO         ;; ADJUST THE INDEX SO THAT IT WILL
        ASL     RO         ;; WORK FOR THE ERROR TABLE
        ASL     RO
        ASL     RO
        ADD     2*$ERRTB, RO ;; FORM TABLE POINTER
        MOV     (RO)+, 2$   ;; PICKUP "ERROR MESSAGE" POINTER
        BEQ    3$         ;; SKIP TIMEOUT IF NO POINTER
        TYPE   2$         ;; TYPE THE "ERROR MESSAGE"
                           ;; "ERROR MESSAGE" POINTER GOES HERE
2$:      .WORD   0         ;; "CARRIAGE RETURN" & "LINE FEED"
        TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
        MOV    (RO)+, 4$   ;; PICKUP "DATA HEADER" POINTER
        BEQ    5$         ;; SKIP TIMEOUT IF 0
        TYPE   4$         ;; TYPE THE "DATA HEADER"
                           ;; "DATA HEADER" POINTER GOES HERE
3$:      .WORD   0         ;; "CARRIAGE RETURN" & "LINE FEED"
        TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
        MOV    R1, -(SP)  ;; SAVE R1
        MOV    (RO)+, R1  ;; PICKUP "DATA TABLE" POINTER
        BEQ    9$         ;; BR IF NO DATA TO BE TYPED
        MOV    (RO)+, RO  ;; PICKUP "DATA FORMAT" POINTER
        TSTB  (RO)+      ;; "OCTAL" OR "DECIMAL"
        BNE   7$         ;; BR IF DECIMAL
        MOV    2(R1)+, -(SP)
                           ;; SAVE 2(R1)+ FOR TIMEOUT
        TYPDC  8$         ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
        BR    8$
7$:      MOV     2(R1)+, -(SP)
                           ;; SAVE 2(R1)+ FOR TIMEOUT
        TYPDS  9$         ;; GO TYPE--DECIMAL ASCII WITH SIGN
        TST   (R1)       ;; IS THERE ANOTHER NUMBER?
        BEQ   9$         ;; BR IF NO
        TYPE  11$        ;; TYPE TWO(2) SPACES
        BR    6$         ;; LOOP
8$:      MOV     (SP)+, R1  ;; RESTORE R1
        MOV     (SP)+, RO  ;; RESTORE RO
        TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
9$:      MOV     (SP)+, R1  ;; RESTORE R1
        MOV     (SP)+, RO  ;; RESTORE RO
        TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
10$:     MOV     (SP)+, R1  ;; RESTORE R1
        MOV     (SP)+, RO  ;; RESTORE RO
        TYPE   $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"

```

E13

```

8813 057422 000207
8814 057424 020040 000
8815 057430
8816
8817
8818
8819
8820
8821
8822
8823
8824
8825
8826
8827
8828
8829
8830
8831
8832
8833 057430 105737 001157
8834 057434 100002
8835 057436 000000
8836 057440 000407
8837 057442 010046
8838 057444 017600 000002
8839 057450 112046
8840 057452 001005
8841 057454 005726
8842 057456 012600
8843 057460 062716 000002
8844 057464 000002
8845 057466 122716 000011
8846 057472 001430
8847 057474 122716 000200
8848 057500 001006
8849 057502 005726
8850 057504 104401
8851 057506 001207
8852 057510 105037 057644
8853 057514 000755
8854 057516 004737 057600
8855 057522 123726 001156
8856 057526 001350
8857 057530 013746 001154
8858
8859 057534 105366 000001
8860 057540 002770
8861 057542 004737 057600
8862 057546 105337 057644
8863 057552 000770
8864
8865
8866
8867 057554 112716 000040
8868 057560 004737 057600

RIS PC
ASCIZ / /
EVEN
.SBTTL TYPE ROUTINE

*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
*
*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*
$TYPE: TSTB $TFPLG ;; IS THERE A TERMINAL?
BPL 1$ ;; BR IF YES
HALT ;; HALT HERE IF NO TERMINAL
BR 3$ ;; LEAVE
1$: MOV RO,-(SP) ;; SAVE RO
MOV 02(SP),RO ;; GET ADDRESS OF ASCIZ STRING
2$: MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+,RO ;; RESTORE RO
3$: ADD #2,(SP) ;; ADJUST RETURN PC
RTI ;; RETURN
4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
BEQ 8$
CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
BNE 5$
TST (SP)+ ;; POP <CR><LF> EQUIV
TYPE $CRLF ;; TYPE A CR AND LF
5$: CLR B $CHARCNT ;; CLEAR CHARACTER COUNT
BR 2$ ;; GET NEXT CHARACTER
5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
BNE 2$ ;; IF NO GO GET NEXT CHAR.
MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
JSR PC,$TYPEC ;; GO TYPE A NULL
DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
BR 7$ ;; LOOP

;HORIZONTAL TAB PROCESSOR
8$: MOVB #' (SP) ;; REPLACE TAB WITH SPACE
9$: JSR PC,$TYPEC ;; TYPE A SPACE

```

```

8869 057564 132737 000007 057644 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
8870 057572 001372 BNE 9$ ;;TAB STOP
8871 057574 005726 TST (SP)+ ;;POP SPACE OFF STACK
8872 057576 000724 BR 2$ ;;GET NEXT CHARACTER
8873 057600 105777 121344 $TYPEPC: TSTB 2$TPS ;;WAIT UNTIL PRINTER IS READY
8874 057604 100375 BPL $TYPEPC
8875 057606 116677 000002 121336 MOVB 2(SP),2$TPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
8876 057614 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
8877 057622 001003 BNE 1$ ;;BRANCH IF NO
8878 057624 105037 057644 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
8879 057630 000406 BR $TYPEX ;;EXIT
8880 057632 122766 000012 000002 1$: CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?
8881 057640 001402 BEQ $TYPEX ;;BRANCH IF YES
8882 057642 105227 INCB (PC)+ ;;COUNT THE CHARACTER
8883 057644 000000 $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
8884 057646 000207 $TYPEX: RTS PC
8885
8886 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
8887
8888 *****
8889 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
8890 *OCTAL (ASCII) NUMBER AND TYPE IT.
8891 *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
8892 *CALL:
8893 * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
8894 * TYPOS ;;CALL FOR TYPEOUT
8895 * .BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
8896 * .BYTE M ;;M=1 OR 0
8897 * ;;1=TYPE LEADING ZEROS
8898 * ;;0=SUPPRESS LEADING ZEROS
8899
8900 *$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
8901 *$TYPOS OR $TYPOC
8902 *CALL:
8903 * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
8904 * TYPON ;;CALL FOR TYPEOUT
8905
8906 *$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
8907 *CALL:
8908 * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
8909 * TYPOC ;;CALL FOR TYPEOUT
8910
8911 057650 017646 000000 060073 $TYPOS: MOV 2(SP),-(SP) ;;PICKUP THE MODE
8912 057654 116637 000001 060073 MOVB 1(SP),$OFILL ;;LOAD ZERO FILL SWITCH
8913 057662 112637 060075 MOVB (SP)+,$OMODE+1 ;;NUMBER OF DIGITS TO TYPE
8914 057666 062716 000002 ADD #2,(SP) ;;ADJUST RETURN ADDRESS
8915 057672 000406 BR $TYPON
8916 057674 112737 000001 060073 $TYPOC: MOVB #1,$OFILL ;;SET THE ZERO FILL SWITCH
8917 057702 112737 000006 060075 MOVB #6,$OMODE+1 ;;SET FOR SIX(6) DIGITS
8918 057710 112737 000005 060072 $TYPON: MCVB #5,$OCNT ;;SET THE ITERATION COUNT
8919 057716 010346 MOV R3,-(SP) ;;SAVE R3
8920 057720 010446 MOV R4,-(SP) ;;SAVE R4
8921 057722 010546 MOV R5,-(SP) ;;SAVE R5
8922 057724 113704 060075 MOVB $OMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
8923 057730 005404 NEG R4
8924 057732 062704 000006 ADD #6,R4 ;;SUBTRACT IT FOR MAX. ALLOWED
    
```

G13

8925	057736	110437	060074		MOV B	R4, \$OMODE	;; SAVE IT FOR USE
8926	057742	113704	060073		MOV B	\$OFILL, R4	;; GET THE ZERO FILL SWITCH
8927	057746	016605	000012		MOV	12(SP), R5	;; PICKUP THE INPUT NUMBER
8928	057752	005003			CLR	R3	;; CLEAR THE OUTPUT WORD
8929	057754	006105		1\$:	ROL	R5	;; ROTATE MSB INTO "C"
8930	057756	000404			BR	3\$;; GO DO MSB
8931	057760	006105		2\$:	ROL	R5	;; FORM THIS DIGIT
8932	057762	006105			ROL	R5	
8933	057764	006105			ROL	R5	
8934	057766	010503			MOV	R5, R3	
8935	057770	006103		3\$:	ROL	R3	;; GET LSB OF THIS DIGIT
8936	057772	105337	060074		DECB	\$OMODE	;; TYPE THIS DIGIT?
8937	057776	100016			SPL	7\$;; BR IF NO
8938	060000	042703	177770		BIC	#177770, R3	;; GET RID OF JUNK
8939	060004	001002			BNE	4\$;; TEST FOR 0
8940	060006	005704			TST	R4	;; SUPPRESS THIS 0?
8941	060010	001403			BEG	5\$;; BR IF YES
8942	060012	005204		4\$:	INC	R4	;; DON'T SUPPRESS ANYMORE 0'S
8943	060014	052703	000060		BIS	#'0, R3	;; MAKE THIS DIGIT ASCII
8944	060020	052703	000040	5\$:	BIS	#' , R3	;; MAKE ASCII IF NOT ALREADY
8945	060024	110337	060070		MOV B	R3, 8\$;; SAVE FOR TYPING
8946	060030	104401	060070		TYPE	8\$;; GO TYPE THIS DIGIT
8947	060034	105337	060072	7\$:	DECB	\$OCNT	;; COUNT BY 1
8948	060040	003347			BGT	2\$;; BR IF MORE TO DO
8949	060042	002402			BLT	6\$;; BR IF DONE
8950	060044	005204			INC	R4	;; INSURE LAST DIGIT ISN'T A BLANK
8951	060046	000744			BR	2\$;; GO DO THE LAST DIGIT
8952	060050	012605		6\$:	MOV	(SP)+, R5	;; RESTORE R5
8953	060052	012604			MOV	(SP)+, R4	;; RESTORE R4
8954	060054	012603			MOV	(SP)+, R3	;; RESTORE R3
8955	060056	016666	000002 000004		MOV	2(SP), 4(SP)	;; SET THE STACK FOR RETURNING
8956	060064	012616			MOV	(SP)+, (SP)	
8957	060066	000002			RTI		;; RETURN
8958	060070	000		8\$:	.BYTE	0	;; STORAGE FOR ASCII DIGIT
8959	060071	000			.BYTE	0	;; TERMINATOR FOR TYPE ROUTINE
8960	060072	000		\$OCNT:	.BYTE	0	;; OCTAL DIGIT COUNTER
8961	060073	000		\$OFILL:	.BYTE	0	;; ZERO FILL SWITCH
8962	060074	000000		\$OMODE:	.WORD	0	;; NUMBER OF DIGITS TO TYPE
8963				.SBTTL	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE		
8964							
8965							*****
8966							*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
8967							*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
8968							*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
8969							*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
8970							*REPLACED WITH SPACES.
8971							*CALL:
8972					*	MOV	NUM, -(SP) ; ; PUT THE BINARY NUMBER ON THE STACK
8973					*	TYPDS	;; GO TO THE ROUTINE
8974							
8975	060076			\$TYPDS:			
8976	060076	010046			MOV	R0, -(SP)	;; PUSH R0 ON STACK
8977	060100	010146			MOV	R1, -(SP)	;; PUSH R1 ON STACK
8978	060102	010246			MOV	R2, -(SP)	;; PUSH R2 ON STACK
8979	060104	010346			MOV	R3, -(SP)	;; PUSH R3 ON STACK
8980	060106	010546			MOV	R5, -(SP)	;; PUSH R5 ON STACK

H13

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 163
DZRJEA.CMB 02-NOV-76 18:40 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```
8981 060110 012746 020200      MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
8982 060114 016605 000020      MOV      20(SP),R5      ;;GET THE INPUT NUMBER
8983 060120 100004          BPL      1$              ;;BR IF INPUT IS POS.
8984 060122 005405          NEG      R5              ;;MAKE THE BINARY NUMBER PCS.
8985 060124 112766 0C0055 000001  MOVB     #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
8986 060132 005000          CLR      R0              ;;ZERO THE CONSTANTS INDEX
8987 060134 012703 060312 1$:      MOV      #SDBLK,R3      ;;SETUP THE OUTPUT POINTER
8988 060140 112723 000040          MOVB     #' ,(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
8989 060144 005002          CLR      R2              ;;CLEAR THE BCD NUMBER
8990 060146 016001 060302          MOV      $DTBL(R0),R1   ;;GET THE CONSTANT
8991 060152 160105          SUB      R1,R5          ;;FORM THIS BCD DIGIT
8992 060154 002402          BLT     4$              ;;BR IF DONE
8993 060156 005202          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
8994 060160 000774          BR      3$              ;;
8995 060162 060105          4$:      ADD     R1,R5          ;;ADD BACK THE CONSTANT
8996 060164 005702          TST     R2              ;;CHECK IF BCD DIGIT=0
8997 060166 001002          BNE     5$              ;;FALL THROUGH IF 0
8998 060170 105716          TSTB    (SP)           ;;STILL DOING LEADING 0'S?
8999 060172 100407          BMI     7$              ;;BR IF YES
9000 060174 176316          5$:      ASLB    (SP)           ;;MSD?
9001 060176 103003          BCC     6$              ;;BR IF NO
9002 060200 116663 000001 177777  MOVB     1(SP),-1(R3)   ;;YES--SET THE SIGN
9003 060206 052702 000060 6$:      BIS     #'0,R2         ;;MAKE THE BCD DIGIT ASCII
9004 060212 052702 000040 7$:      BIS     #' ,R2         ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9005 060216 110223          MOVB     R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9006 060220 005720          TST     (R0)+          ;;JUST INCREMENTING
9007 060222 020027 000010          CMP     R0,#10        ;;CHECK THE TABLE INDEX
9008 060226 002746          BLT     2$              ;;GO DO THE NEXT DIGIT
9009 060230 003002          BGT     8$              ;;GO TO EXIT
9010 060232 010502          MOV     R5,R2          ;;GET THE LSD
9011 060234 000764          BR      6$              ;;GO CHANGE TO ASCII
9012 060236 105726          8$:      TSTB    (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
9013 060240 100003          BPL     9$              ;;BR IF NO
9014 060242 116663 177777 177776  MOVB     -1(SP),-2(R3)  ;;YES--SET THE SIGN FOR TYPING
9015 060250 105013          9$:      CLRB    (R3)         ;;SET THE TERMINATOR
9016 060252 012605          MOV     (SP)+,R5       ;;POP STACK INTO R5
9017 060254 012603          MOV     (SP)+,R3       ;;POP STACK INTO R3
9018 060256 012602          MOV     (SP)+,R2       ;;POP STACK INTO R2
9019 060260 012601          MOV     (SP)+,R1       ;;POP STACK INTO R1
9020 060262 012600          MOV     (SP)+,R0       ;;POP STACK INTO R0
9021 060264 104401 060312          TYPE    $SDBLK        ;;NOW TYPE THE NUMBER
9022 060270 016666 000002 000004  MOV     2(SP),4(SP)    ;;ADJUST THE STACK
9023 060276 012616          MOV     (SP)+,(SP)
9024 060300 000002          RTI
9025 060302 023420          $DTBL: 10000.          ;;RETURN TO USER
9026 060304 001750          1000.
9027 060306 000144          100.
9028 060310 000012          10.
9029 060312 000004          $SDBLK: .BLKW 4
9030          .SBTTL TTY INPUT ROUTINE
9031
9032          ;:*****
9033          .ENABL LSB
9034 060322 000000          $TKCNT: .WORD 0      ;;NUMBER OF ITEMS IN QUEUE
9035 060324 000000          $TKQIN: .WORD 0     ;;INPUT POINTER
9036 060326 000000          $TKQOUT: .WORD 0    ;;OUTPUT POINTER
```

```

9037 060330 000001      $TKQSRT: .BLKB 1          ;;TTY KEYBOARD QUEUE
9038      060331      $TKQEND=.
9039      060332      .EVEN
9040
9041      ;*TK INITIALIZE ROUTINE
9042      ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
9043      ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
9044
9045      ;*CALL:
9046      ;*      JSR      PC,$TKINT
9047      ;*      RETURN
9048
9049 060332 005037 060322 $TKINT: CLR      $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
9050 060336 012737 060330 060324 MOV      $TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
9051 060344 013737 060324 060326 MOV      $TKQIN,$TKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
9052 060352 012737 060402 000060 MOV      $TKSRV,$TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
9053 060360 012737 000200 000062 MOV      #200,$TKVEC+2 ;;"BR" LEVEL 4
9054 060366 005777 120554 TST      $TKB          ;;CLEAR DONE FLAG
9055 060372 012777 000100 120544 MOV      #100,$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
9056 060400 000207      RTS      PC          ;;RETURN TO CALLER
9057
9058      ;*TK SERVICE ROUTINE
9059      ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
9060      ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
9061      ;*IT IN THE QUEUE.
9062
9063 060402 117746 120540 $TKSRV: MOVB     $TKB,-(SP) ;;PICKUP THE CHARACTER
9064 060406 042716 177600 BIC      #1C177,(SP) ;;STRIP THE JUNK
9065 060412 021627 000007 1$: CMP      (SP),#7 ;;IS IT A CONTROL G?
9066 060416 001004 BNE      2$ ;;BRANCH IF NO
9067 060420 022737 000176 001140 CMP      $SWREG,$SWR ;;IS SOFT-SWR SELECTED?
9068 060426 001500 BEQ      6$ ;;GO TO SWR CHANGE
9069
9070      2$:
9071 060430 022737 000001 060322 CMP      #1,$TKCNT ;;IS THE QUEUE FULL?
9072 060436 001004 BNE      3$ ;;BRANCH IF NO
9073 060440 104401 001202 TYPE     $BELL ;;RING THE TTY BELL
9074 060444 005726 TST      (SP)+ ;;CLEAN CHARACTER OFF OF STACK
9075 060446 000451 BR       5$ ;;EXIT
9076 060450 021627 000023 3$: CMP      (SP),#23 ;;IS IT A CONTROL-S?
9077 060454 001021 BNE      32$ ;;BRANCH IF NO
9078 060456 005077 120462 CLR      $TKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
9079 060462 005726 TST      (SP)+ ;;CLEAN CHAR OFF STACK
9080 060464 105777 120454 31$: TSTB     $TKS      ;;WAIT FOR A CHAR
9081 060470 100375 BPL      31$ ;;LOOP UNTIL ITS THERE
9082 060472 117746 120450 MOVB     $TKB,-(SP) ;;GET THE CHARACTER
9083 060476 042716 177600 BIC      #1C177,(SP) ;;MAKE IT 7-BIT ASCII
9084 060502 022627 000021 CMP      (SP)+,#21 ;;IS IT A CONTROL-Q?
9085 060506 001366 BNE      31$ ;;BRANCH IF NO
9086 060510 012777 000100 120426 MOV      #100,$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
9087 060516 000002 RTI      ;;RETURN
9088 060520 005237 060322 32$: INC      $TKCNT ;;COUNT THIS CHARACTER
9089 060524 021627 000140 CMP      (SP),#140 ;;IS IT UPPER CASE?
9090 060530 002405 BLT      4$ ;;BRANCH IF YES
9091 060532 021627 000175 CMP      (SP),#175 ;;IS IT A SPECIAL CHAR?
9092 060536 003002 BGT      4$ ;;BRANCH IF YES

```

```

9093 060540 042716 000040          BIC      #40,(SP)      ;;MAKE IT UPPER CASE
9094 060544 :12677 177554      4$:  MOVB   (SP)+,@STKQIN  ;;AND PUT IT IN QUEUE
9095 060550 005237 060324          INC     STKQIN        ;;UPDATE THE POINTER
9096 060554 023727 060324 060331    CMP    STKQIN,#STKQEND ;;GO OFF THE END?
9097 060562 001003          BNE     5$           ;;BRANCH IF NO
9098 060564 012737 060330 060324    MOV    #STKQSRST,STKQIN ;;RESET THE POINTER
9099 060572 000002      5$:  RTI                ;;RETURN
9100
9101          ;;*****
9102          ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
9103          ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
9104          ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
9105          ;;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
9106 060574 022737 000176 001140    $CKSWR: CMP    #SWREG,SWR    ;;IS THE SOFT-SWR SELECTED
9107 060602 001104          BNE     15$         ;;EXIT IF NOT
9108 060604 105777 120334          TSTB   @STKS        ;;IS A CHAR WAITING?
9109 060610 100101          BPL     15$         ;;IF NOT, EXIT
9110 060612 117746 120330          MOVB   @STKB,-(SP)   ;;YES
9111 060616 042716 177600          BIC    #177,(SP)    ;;MAKE IT 7-BIT ASCII
9112 060622 021627 000007          CMP    (SP),#7      ;;IS IT A CONTROL-G?
9113 060626 001300          BNE     2$           ;;IF NOT, PUT IT IN THE TTY QUEUE
9114          ;;AND EXIT
9115
9116          ;;*****
9117          ;;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
9118          ;;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
9119          ;;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
9120 060630 123727 001134 000001    6$:  CMPB   $AUTOB,#1   ;;ARE WE RUNNING IN AUTO-MODE?
9121 060636 001674          BEQ     2$           ;;BRANCH IF YES
9122 060640 005726          TST    (SP)+        ;;CLEAR CONTROL-G OFF STACK
9123 060642 004737 060332          JSR    PC,STKINT    ;;FLUSH THE TTY INPUT QUEUE
9124 060646 005077 120272          CLR    @STKS        ;;DISABLE TTY KEYBOARD INTERRUPTS
9125 060652 112737 000001 001135    MOVB   #1,$INTAG    ;;SET INTERRUPT MODE INDICATOR
9126
9127 060660 104401 061436          TYPE   , $CNTLG     ;;ECHO THE CONTROL-G (1G)
9128 060664 104401 061443          $GTSWR: TYPE   $MSWR  ;;TYPE CURRENT CONTENTS
9129 060670 013746 000176          MOV    $WREG,-(SP)  ;;SAVE SWREG FOR TYPEOUT
9130 060674 104402          TYPOC          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
9131 060676 104401 061454          TYPE   , $MNEW     ;;PROMPT FOR NEW SWR
9132 060702 005046          19$:  CLR    -(SP)       ;;CLEAR COUNTER
9133 060704 005046          CLR    -(SP)       ;;THE NEW SWR
9134 060706 105777 120232          7$:  TSTB   @STKS        ;;CHAR THERE?
9135 060712 100375          BPL     7$         ;;IF NOT TRY AGAIN
9136
9137 060714 117746 120226          MOVB   @STKB,-(SP)  ;;PICK UP CHAR
9138 060720 042716 177600          BIC    #177,(SP)   ;;MAKE IT 7-BIT ASCII
9139
9140
9141
9142 060724 021627 000025          9$:  CMP    (SP),#25    ;;IS IT A CONTROL-U?
9143 060730 001005          BNE     10$        ;;BRANCH IF NOT
9144 060732 104401 061431          TYPE   , $CNTLU    ;;YES, ECHO CONTROL-U (1U)
9145 060736 062706 000006          20$:  ADD    #6,SP      ;;IGNORE PREVIOUS INPUT
9146 060742 000757          BR     19$         ;;LET'S TRY IT AGAIN
9147
9148

```

K13

```

9149 060744 021627 000015 10$: CMP (SP),#15 ;: IS IT A <CR>?
9150 060750 001022 BNE 16$ ;: BRANCH IF NO
9151 060752 005766 000004 TST 4(SP) ;: YES, IS IT THE FIRST CHAR?
9152 060756 001403 BEQ 11$ ;: BRANCH IF YES
9153 060760 016677 000002 120152 MOV 2(SP),@SWR ;: SAVE NEW SWR
9154 060766 062706 000006 11$: ADD #6,SP ;: CLEAR UP STACK
9155 060772 104401 001207 14$: TYPE $CRLF ;: ECHO <CR> AND <LF>
9156 060776 123727 001135 000001 CMPB $INTAG,#1 ;: RE-ENABLE TTY KBD INTERRUPTS?
9157 061004 001003 BNE 15$ ;: BRANCH IF NOT
9158 061006 012777 000100 120130 MOV #100,@$TKS ;: RE-ENABLE TTY KBD INTERRUPTS
9159 061014 000002 15$: RTI ;: RETURN
9160 061016 004737 057600 16$: JSR PC,$TYPEC ;: ECHO CHAR
9161 061022 021627 000060 CMP (SP),#60 ;: CHAR < 0?
9162 061026 002420 BLT 18$ ;: BRANCH IF YES
9163 061030 021627 000067 CMP (SP),#67 ;: CHAR > 7?
9164 061034 003015 BGT 18$ ;: BRANCH IF YES
9165 061036 042726 000060 BIC #60,(SP)+ ;: STRIP-OFF ASCII
9166 061042 005766 000002 TST 2(SP) ;: IS THIS THE FIRST CHAR
9167 061046 001403 BEQ 17$ ;: BRANCH IF YES
9168 061050 006316 ASL (SP) ;: NO, SHIFT PRESENT
9169 061052 006316 ASL (SP) ;: CHAR OVER TO MAKE
9170 061054 006316 ASL (SP) ;: ROOM FOR NEW ONE.
9171 061056 005266 000002 17$: INC 2(SP) ;: KEEP COUNT OF CHAR
9172 061062 056616 177776 BIS -2(SP),(SP) ;: SET IN NEW CHAR
9173 061066 000707 BR 7$ ;: GET THE NEXT ONE
9174 061070 104401 001206 18$: TYPE $QUES ;: TYPE ?<CR><LF>
9175 061074 000720 BR 20$ ;: SIMULATE CONTROL-U
9176 .DSABL LSB
9177
9178
9179 ;:*****
9180 ;:THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9181 ;:CALL:
9182 ;* RDCHR ;: GET A CHARACTER FROM THE QUEUE
9183 ;* RETURN HERE ;: CHARACTER IS ON THE STACK
9184 ;* ;: WITH PARITY BIT STRIPPED OFF
9185 ;:
9186
9187 061076 011646 $RDCHR: MOV (SP),-(SP) ;: PUSH DOWN THE PC AND
9188 061100 016666 000004 000002 MOV 4(SP),2(SP) ;: THE PS
9189 061106 005066 000004 CLR 4(SP) ;: GET READY FOR A CHARACTER
9190 061112 005046 CLR -(SP) ;: PUT NEW PS ON STACK
9191 061114 012746 061122 MOV #64$,-(SP) ;: PUT NEW PC ON STACK
9192 061120 000002 RTI ;: POP NEW PC AND PS
9193 061122 64$:
9194 061122 005737 060322 1$: TST $TKCNT ;: WAIT ON A CHARACTER
9195 061126 001775 BEQ 1$
9196 061130 005337 060322 DEC $TKCNT ;: DECREMENT THE COUNTER
9197 061134 117766 177166 000004 MOVB @$TKQOUT,4(SP) ;: GET ONE CHARACTER
9198 061142 005237 060326 INC $TKQOUT ;: UPDATE THE POINTER
9199 061146 023727 060326 060331 CMP $TKQOUT,$$TKQEND ;: DID IT GO OFF OF THE END?
9200 061154 001003 BNE 2$ ;: BRANCH IF NO
9201 061156 012737 060330 060326 MOV $$TKQ$RT,$$TKQOUT ;: RESET THE POINTER
9202 061164 000002 2$: RTI ;: RETURN
9203 ;:*****
9204 ;:THIS ROUTINE WILL INPUT A STRING FROM THE TTY

```



```

9205      ;*CALL:
9206      ;*      RDLIN      ;: INPUT A STRING FROM THE TTY
9207      ;*      RETURN HERE ;: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
9208      ;*                               ;: TERMINATOR WILL BE A BYTE OF ALL 0'S
9209
9210      061166 010346      $RDLIN: MOV      R3-(SP)      ;: SAVE R3
9211      061170 005046      CLR      -(SP)      ;: CLEAR THE RUBOUT KEY
9212      061172 012703 061422 1$:      MOV      $TTYIN,R3      ;: GET ADDRESS
9213      061176 022703 061431 2$:      CMP      $TTYIN+7,R3      ;: BUFFER FULL?
9214      061202 101456      BLOS     4$          ;: BR IF YES
9215      061204 104410      RDCHR           ;: GO READ ONE CHARACTER FROM THE TTY
9216      061206 112613      MOVB     (SP)+,(R3)      ;: GET CHARACTER
9217      061210 122713 000177 10$:     CMPB     #177,(R3)      ;: IS IT A RUBOUT
9218      061214 001022      BNE      5$          ;: BR IF NO
9219      061216 005716      TST     (SP)          ;: IS THIS THE FIRST RUBOUT?
9220      061220 001007      BNE      6$          ;: BR IF NO
9221      061222 112737 000134 061420      MOVB     #' \ ,9$      ;: TYPE A BACK SLASH
9222      061230 104401 061420      TYPE     ,9$          ;:
9223      061234 012716 177777      MOV      #-1,(SP)      ;: SET THE RUBOUT KEY
9224      061240 005303      DEC      R3          ;: BACKUP BY ONE
9225      061242 020327 061422 6$:      CMP      R3,$TTYIN      ;: STACK EMPTY?
9226      061246 103434      BLO      4$          ;: BR IF YES
9227      061250 111337 061420      MOVB     (R3),9$      ;: SETUP TO TYPEOUT THE DELETED CHAR.
9228      061254 104401 061420      TYPE     ,9$          ;: GO TYPE
9229      061260 000746      BR       2$          ;: GO READ ANOTHER CHAR.
9230      061262 005716      5$:      TST     (SP)          ;: RUBOUT KEY SET?
9231      061264 001406      BEQ      7$          ;: BR IF NO
9232      061266 112737 000134 061420      MOVB     #' \ ,9$      ;: TYPE A BACK SLASH
9233      061274 104401 061420      TYPE     ,9$          ;:
9234      061300 005016      CLR      (SP)          ;: CLEAR THE RUBOUT KEY
9235      061302 122713 000025 7$:      CMPB     #25,(R3)      ;: IS CHARACTER A CTRL U?
9236      061306 001003      BNE      8$          ;: BR IF NO
9237      061310 104401 061431      TYPE     $CNTLU      ;: TYPE A CONTROL "U"
9238      061314 000726      BR       1$          ;: GO START OVER
9239      061316 122713 000022 8$:      CMPB     #22,(R3)      ;: IS CHARACTER A "↑R"?
9240      061322 001011      BNE      3$          ;: BRANCH IF NO
9241      061324 105013      CLRB     (R3)          ;: CLEAR THE CHARACTER
9242      061326 104401 001207      TYPE     $CRLF      ;: TYPE A "CR" & "LF"
9243      061332 104401 061422      TYPE     $TTYIN      ;: TYPE THE INPUT STRING
9244      061336 000717      BR       2$          ;: GO PICKUP ANOTHER CHARACTER
9245      061340 104401 001206 4$:      TYPE     $QUES      ;: TYPE A '?'
9246      061344 000712      BR       1$          ;: CLEAR THE BUFFER AND LOOP
9247      061346 111337 061420 3$:      MOVB     (R3),9$      ;: ECHO THE CHARACTER
9248      061352 104401 061420      TYPE     ,9$          ;:
9249      061356 122723 000015      CMPB     #15,(R3)+      ;: CHECK FOR RETURN
9250      061362 001305      BNE      2$          ;: LOOP IF NOT RETURN
9251      061364 105063 177777      CLRB     -1(R3)        ;: CLEAR RETURN (THE 15)
9252      061370 104401 001210      TYPE     $LF          ;: TYPE A LINE FEED
9253      061374 005726      TST     (SP)+          ;: CLEAN RUBOUT KEY FROM THE STACK
9254      061376 012603      MOV      (SP)+,R3      ;: RESTORE R3
9255      061400 011646      MOV      (SP),-(SP)     ;: ADJUST THE STACK AND PUT ADDRESS OF THE
9256      061402 016666 000004 000002      MOV      4(SP),2(SP)    ;: FIRST ASCII CHARACTER ON IT
9257      061410 012766 061422 000004      MOV      $TTYIN,4(SP)  ;:
9258      061416 000002      RTI           ;: RETURN
9259      061420      000      9$:      .BYTE   0          ;: STORAGE FOR ASCII CHAR. TO TYPE
9260      061421      000      .BYTE   0          ;: TERMINATOR
    
```

M13

```

9261 061422 000007 $TTYIN: .BLKB 7 ;;RESERVE 7 BYTES FOR TTY INPUT
9262 061431 136 006525 000012 $CNTLU: .ASCIZ /↑U/<15><12> ;;CONTROL "U"
9263 061436 043536 005015 000 $CNTLG: .ASCIZ /↑G/<15><12> ;;CONTROL "G"
9264 061443 015 051412 051127 $MSWR: .ASCIZ <15><12>/SWR = /
9265 061450 036440 000040
9266 061454 020040 042516 020127 $MNEW: .ASCIZ / NEW = /
9267 061462 020075 000
9268 061466
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283 061466 011646 $RDOCT: MOV (SP),-(SP) ;;PROVIDE SPACE FOR THE
9284 061470 016566 000004 000002 MOV 4(SP),2(SP) ;;INPUT NUMBER
9285 061476 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
9286 061500 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
9287 061502 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
9288 061504 104411 1$: RDLIN ;;READ AN ASCIZ LINE
9289 061506 012600 MOV (SP)+,R0 ;;GET ADDRESS OF 1ST CHARACTER
9290 061510 010037 061614 MOV R0,5$ ;;AND SAVE IT
9291 061514 005001 CLR R1 ;;CLEAR DATA WORD
9292 061516 005002 CLR R2
9293 061520 112046 2$: MOVB (R0)+,-(SP) ;;PICKUP THIS CHARACTER
9294 061522 001420 BEQ 3$ ;;IF ZERO GET OUT
9295 061524 122716 000060 CMPB #'0,(SP) ;;MAKE SURE THIS CHARACTER
9296 061530 003026 BGT 4$ ;;IS AN OCTAL DIGIT
9297 061532 122716 000067 CMPB #'7,(SP)
9298 061536 002423 BLT 4$
9299 061540 006301 ASL R1 ;;*2
9300 061542 006102 ROL R2
9301 061544 006301 ASL R1 ;;*4
9302 061546 006102 ROL R2
9303 061550 006301 ASL R1 ;;*8
9304 061552 006102 ROL R2
9305 061554 042716 177770 BIC #'C7,(SP) ;;STRIP THE ASCII JUNK
9306 061560 062601 ADD (SP)+,R1 ;;ADD IN THIS DIGIT
9307 061562 000756 BR 2$ ;;LOOP
9308 061564 005726 3$: TST (SP)+ ;;CLEAN TERMINATOR FROM STACK
9309 061566 010166 000012 MOV R1,12(SP) ;;SAVE THE RESULT
9310 061572 010237 061624 MOV R2,$HIOCT
9311 061576 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
9312 061600 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
9313 061602 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
9314 061604 000002 RTI ;;RETURN
9315 061606 005726 4$: TST (SP)+ ;;CLEAN PARTIAL FROM STACK
9316 061610 !05010 CLR (R0) ;;SET A TERMINATOR

```

N13

MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 169
DZRJEA.CMB 02-NOV-76 18:40 READ AN OCTAL NUMBER FROM THE TTY

9317 061612 104401
9318 061614 000000
9319 061616 104401 001206
9320 061622 000730
9321 061624 000000
9322
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339 061626
9340 061626 010046
9341 061630 010146
9342 061632 010246
9343 061634 010346
9344 061636 010446
9345 061640 010546
9346 061642 016646 000022
9347 061646 016646 000022
9348 061652 016646 000022
9349 061656 016646 000022
9350 061662 000002
9351
9352
9353
9354
9355 061664
9356 061664 012666 000022
9357 061670 012666 000022
9358 061674 012666 000022
9359 061700 012666 000022
9360 061704 012605
9361 061706 012604
9362 061710 012603
9363 061712 012602
9364 061714 012601
9365 061716 012600
9366 061720 000002
9367
9368
9369
9370
9371
9372

;;TYPE UP THRU THE BAD CHAR.
SS: .WORD 0
TYPE \$QUES ;;"?" "CR" & "LF"
BR 1\$;:TRY AGAIN
\$HI OCT: .WORD 0 ;:HIGH ORDER BITS GO HERE
.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

*SAVE RO-R5
*CALL:
* SAVPEC
*UPON RETURN FROM \$SAVREG THE STACK WILL LOOK LIKE:
*
*TOP---(+16)
* +2---(+18)
* +4---R5
* +6---R4
* +8---R3
*+10---R2
*+12---R1
*+14---R0

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

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

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

```

9373
9374
9375
9376 061722 010046
9377 061724 016600 000002
9378 061730 005740
9379 061732 111000
9380 061734 006300
9381 061736 016000 061756
9382 061742 000200
9383
9384
9385
9386 061744 011646
9387 061746 016666 000004 000002
9388 061754 000002
9389
9390
9391
9392
9393
9394
9395
9396
9397 061756 061744
9398 061760 057430
9399 061762 057674
9400 061764 057650
9401 061766 057710
9402 061770 060076
9403
9404 061772 060664
9405
9406 061774 060574
9407 061776 061076
9408 062000 061166
9409 062002 061466
9410 062004 061626
9411 062006 061664
9412
9413
9414
9415
9416
9417
9418
9419 062010 005015 046412 044501
9420 062016 042116 041505 030455
9421 062024 026461 055104 045122
9422 062032 026505 006501 012
9423 062037 122 030120 027464
9424 062044 027465 020066 052504
9425 062052 046101 041440 047117
9426 062060 051124 046117 042514
9427 062066 020122 047514 044507
9428 062074 020103 042524 052123

```

;*GO TO THAT ROUTINE.

```

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

```

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

```

STRAP2: MOV (SP), -(SP)      ;; MOVE THE PC DOWN
        MOV 4(SP), 2(SP)     ;; MOVE THE PSW DOWN
        RTI                  ;; RESTORE THE PSW

```

.SBTTL TRAP TABLE

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

```

ROUTINE
-----
STRPAD: .WORD STRAP2
        $TYPE      ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
        $TYPOC     ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS     ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON     ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS     ;;CALL=TYPDS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
        $GTSWR    ;;CALL=GTSWR     TRAP+6(104406) GET SOFT-SWR SETTING
        $CKSWR    ;;CALL=CKSWR     TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
        $RDCHR    ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
        $RDLIN    ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        $RDOCT    ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
        $SAVREG   ;;CALL=SAVREG    TRAP+13(104413) SAVE R0-R5 ROUTINE
        $RESREG   ;;CALL=RESREG    TRAP+14(104414) RESTORE R0-R5 ROUTINE

```

;;*****

.SBTTL TELETYPE MESSAGES

;;*****

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

.ASCII 2RPO4/5/6 DUAL CONTROLLER LOGIC TEST - PART 12<CR><LF><LF>

062102 026440 050040 051101
062110 020124 006461 005012
062116 000
062117 015 042412 052116
062124 051105 042040 044522
062132 042526 040440 042104
062140 042522 051523 020072
062146 000
062147 111 053116 046101
062154 042111 040440 042104
062162 042522 051523 005015
062170 000
062171 015 050012 051117
062176 020124 020101 042101
062204 051104 051505 020123
062212 051511 020072 000
062217 015 050012 051117
062224 020124 020102 042101
062232 051104 051505 020123
062240 051511 020072 000
062245 015 051412 051531
062252 042524 020115 052515
062260 052123 044040 053101
062266 020105 046047 020047
062274 051117 023440 023520
062302 041440 047514 045503
062310 005015 000012
062314 042412 052116 051105
062322 052040 051505 020124
062330 035043 000040
062334 047111 040526 044514
062342 020104 042524 052123
062350 047040 046525 042502
062356 006522 000012
062362 042440 051122 051117
062370 006523 000012
062374 005015 052012 042510
062402 050040 042522 042523
062410 052116 040440 042104
062416 042522 051523 047440
062424 020106 044124 020105
062432 044122 030461 024040
062440 050122 051503 024461
062446 044440 035123 000040
062454 042412 052116 051105
062462 047040 053505 051040
062470 030510 020061 042101
062476 051104 051505 035123
062504 000040

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: /

NOCLKCK: .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 (RPCS1) IS: /

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

;;*****

.SBTTL TEST ERROR MESSAGES

;;*****

9485	062506	051127	047117	020107	EM1:	.ASCIZ /WRONG DRIVE TYPE/
9486	062514	051104	053111	020105		
9487	062522	054524	042520	000		
9488						
9489	062527	104	044522	042526	EM2:	.ASCIZ /DRIVE NOT ON LINE/
9490	062534	047040	052117	047440		
9491	062542	020116	044514	042516		
9492	062550	000				
9493						
9494	062551	123	051105	040511	EM3:	.ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
9495	062556	020114	052516	041115		
9496	062564	051105	051040	040505		
9497	062572	020104	044124	047522		
9498	062600	043525	020110	040505		
9499	062606	044103	050040	051117		
9500	062614	020124	047516	020124		
9501	062622	044124	020105	040523		
9502	062630	042515	000			
9503						
9504	062633	104	044522	042526	EM4:	.ASCIZ /DRIVE NOT SEIZED BY PORT/
9505	062640	047040	052117	051440		
9506	062646	044505	042532	020104		
9507	062654	054502	050040	051117		
9508	062662	000124				
9509						
9510	062664	051127	047117	020107	EM5:	.ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
9511	062672	052123	052101	051525		
9512	062700	051440	042505	020116		
9513	062706	054502	052040	042510		
9514	062714	051440	044505	044532		
9515	062722	043516	050040	051117		
9516	062730	000124				
9517						
9518	062732	042522	044507	052123	EM6:	.ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
9519	062740	051105	041440	047117		
9520	062746	042524	052116	020123		
9521	062754	042527	042522	051440		
9522	062762	042505	020116	054502		
9523	062770	047440	050120	051517		
9524	062776	052111	020105	047520		
9525	063004	052122	026440	042040		
9526	063012	044522	042526	053440		
9527	063020	051501	051440	044505		
9528	063026	042532	000104			
9529						
9530	063032	042522	044507	052123	EM7:	.ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
9531	063040	051105	041440	047117		
9532	063046	042524	052116	020123		
9533	063054	051127	047117	020107		
9534	063062	043101	042524	020122		
9535	063070	042522	042514	051501		
9536	063076	020105	051117	052040		
9537	063104	046511	047505	052125		
9538	063112	000				
9539						
9540	063113	122	043505	051511	EM10:	.ASCIZ /REGISTER CONTENTS WRONG/

E14

9541	063120	042524	020122	047503	
9542	063126	052116	047105	051524	
9543	063134	053440	047522	043516	
9544	063142	000			
9545					
9546	063143	103	047117	051124	EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
9547	063150	046117	041040	051525	
9548	063156	050040	051101	052111	
9549	063164	020131	051105	047522	
9550	063172	020122	042522	042101	
9551	063200	047111	020107	047111	
9552	063206	044504	040503	042524	
9553	063214	020104	042522	044507	
9554	063222	052123	051105	000	
9555					
9556	063227	104	044522	042526	EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
9557	063234	047040	052117	051440	
9558	063242	044505	042532	020104	
9559	063250	054502	042040	044522	
9560	063256	042526	041440	042514	
9561	063264	051101	041440	046517	
9562	063272	040515	042116	000	
9563					
9564	063277	122	040505	044504	EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
9565	063304	020116	051120	051505	
9566	063312	052105	042040	042517	
9567	063320	020123	047516	020124	
9568	063326	042523	020124	047526	
9569	063334	052514	042515	053040	
9570	063342	046101	042111	043040	
9571	063350	051117	052040	042510	
9572	063356	050040	051117	000124	
9573					
9574	063364	047526	052514	042515	EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
9575	063372	053040	046101	042111	
9576	063400	051440	052105	047440	
9577	063406	020116	044124	020105	
9578	063414	051127	047117	020107	
9579	063422	047520	052122	000	
9580					
9581	063427	101	052124	020116	EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
9582	063434	044502	020124	051127	
9583	063442	047117	020107	043101	
9584	063450	042524	020122	044524	
9585	063456	042515	052517	020124	
9586	063464	020055	042522	052521	
9587	063472	051505	020124	047516	
9588	063500	020124	042523	060124	
9589					
9590	063506	052101	047124	041040	EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
9591	063514	052111	053440	047522	
9592	063522	043516	040440	052106	
9593	063530	051105	051040	046105	
9594	063536	040505	042523	026440	
9595	063544	051040	050505	042525	
9596	063552	052123	051440	052105	

9597	063560	000			
9598					
9599	063561	101	052124	020116	EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
9600	063566	044502	020124	051127	
9601	063574	047117	020107	043101	
9602	063602	042524	020122	042522	
9603	063610	042514	051501	020105	
9604	063616	020055	042522	052521	
9605	063624	051505	020124	047516	
9606	063632	020124	042523	000124	
9607					
9608	063640	051104	053111	020105	EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
9609	063646	047516	020124	042523	
9610	063654	055111	042105	053440	
9611	063662	042510	020116	052101	
9612	063670	047124	041040	052111	
9613	063676	043040	051117	050040	
9614	063704	051117	020124	046103	
9615	063712	040505	042522	000104	
9616					
9617	063720	051104	053111	020105	EM21: .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
9618	063726	042523	055111	042105	
9619	063734	053440	042510	020116	
9620	063742	042532	047522	053440	
9621	063750	044522	052124	047105	
9622	063756	044440	020116	052101	
9623	063764	047124	041040	052111	
9624	063772	000			
9625					
9626	063773	104	044522	042526	EM22: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
9627	064000	047040	052117	044440	
9628	064006	020116	042516	052125	
9629	064014	040522	020114	043101	
9630	064022	042524	020122	044524	
9631	064030	042515	052517	020124	
9632	064036	020055	042522	052521	
9633	064044	051505	020124	047516	
9634	064052	020124	042523	000124	
9635					
9636	064060	044524	042515	052517	EM23: .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
9637	064066	020124	046103	040505	
9638	064074	042522	020104	044124	
9639	064102	020105	051104	053111	
9640	064110	023505	020123	051105	
9641	064116	047522	020122	044502	
9642	064124	000124			
9643					
9644	064126	042522	042514	051501	EM24: .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
9645	064134	020105	047503	046515	
9646	064142	047101	020104	042522	
9647	064150	042514	051501	042105	
9648	064156	042040	044522	042526	
9649	064164	053440	052111	020110	
9650	064172	051105	047522	051522	
9651	064200	051440	052105	000	
9652					

9653	064205	124	046511	047505	EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
9654	064212	052125	047440	042516	
9655	064220	051455	047510	020124	
9656	064226	044504	020104	047516	
9657	064234	020124	042522	051124	
9658	064242	043511	042507	000122	
9659					
9660	064250	051104	053111	020105	EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
9661	064256	047516	020124	047111	
9662	064264	047040	052505	051124	
9663	064272	046101	040440	052106	
9664	064300	051105	051040	046105	
9665	064306	040505	042523	026440	
9666	064314	051040	050505	042525	
9667	064322	052123	047040	052117	
9668	064330	051440	052105	000	
9669					
9670	064335	122	043505	051511	EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
9671	064342	042524	020122	051127	
9672	064350	047117	020107	043101	
9673	064356	042524	020122	042522	
9674	064364	042514	051501	020105	
9675	064372	044527	044124	051040	
9676	064400	050505	042525	052123	
9677	064406	051440	052105	000	
9678					
9679	064413	104	044522	042526	EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
9680	064420	051440	044505	042532	
9681	064426	020104	054502	051040	
9682	064434	046105	040505	042523	
9683	064442	041440	046517	040515	
9684	064450	042116	044440	051523	
9685	064456	042525	020104	044127	
9686	064464	047105	042040	044522	
9687	064472	042526	044440	020116	
9688	064500	042516	052125	040522	
9689	064506	000114			
9690					
9691	064510	051104	053111	020105	EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
9692	064516	047111	047040	052505	
9693	064524	051124	046101	040440	
9694	064532	052106	051105	051040	
9695	064540	046105	040505	042523	
9696	064546	026440	051040	050505	
9697	064554	042525	052123	051440	
9698	064562	052105	000		
9699					
9700	064565	101	052124	020116	EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
9701	064572	044502	020124	051127	
9702	064600	047117	020107	043101	
9703	064606	042524	020122	042522	
9704	064614	040503	044514	051102	
9705	064622	052101	020105	047503	
9706	064630	046515	047101	000104	
9707					
9708	064636	051104	053111	020105	EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/

H14

9709	064644	042522	052524	047122
9710	064652	042105	052040	020117
9711	064660	042516	052125	040522
9712	064666	020114	043111	042040
9713	064674	044522	042526	041440
9714	064702	042514	051101	043440
9715	064710	053111	047105	053440
9716	064716	044510	042514	042040
9717	064724	044522	042526	051440
9718	064732	044505	042532	000104
9719				
9720	064740	051104	053111	020105
9721	064746	042522	052524	047122
9722	064754	042105	052040	020117
9723	064762	042516	052125	040522
9724	064770	020114	043111	046440
9725	064776	051501	041123	051525
9726	065004	044440	044516	020124
9727	065012	044507	042526	020116
9728	065020	044127	046111	020105
9729	065026	051104	053111	020105
9730	065034	042523	055111	042105
9731	065042	000		
9732				
9733	065043	124	046511	047505
9734	065050	052125	047440	042516
9735	065056	051440	047510	020124
9736	065064	044506	042522	020104
9737	065072	044527	044124	052517
9738	065100	020124	042522	044507
9739	065106	052123	051105	040440
9740	065114	041503	051505	000123
9741				
9742	065122	044524	042515	052517
9743	065130	020124	040510	020123
9744	065136	047516	020124	041517
9745	065144	052503	051122	042105
9746	065152	053440	052111	044510
9747	065160	020116	020062	042523
9748	065166	047503	042116	000123
9749				
9750	065174	051104	053111	020105
9751	065202	051511	047040	047117
9752	065210	042455	044530	052123
9753	065216	047105	020124	023450
9754	065224	042516	023504	041040
9755	065232	052111	051440	052105
9756	065240	000051		
9757				
9758	065242	052101	047124	041040
9759	065250	052111	043040	051117
9760	065256	050040	051117	020124
9761	065264	047516	020124	042522
9762	065272	042523	020124	054502
9763	065300	046440	051501	041123
9764	065306	051525	041440	042514

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

EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/

EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/

EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/

EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/

9765	065314	051101	000		
9766					
9767	065317	124	046511	047505	EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
9768	065324	052125	041440	042514	
9769	065332	051101	042105	052040	
9770	065340	042510	040440	052124	
9771	065346	047105	044524	047117	
9772	065354	041040	052111	000	
9773					
9774	065361	104	044522	042526	EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
9775	065366	047040	052117	044440	
9776	065374	020116	042516	052125	
9777	065402	040522	020114	051117	
9778	065410	051440	044505	042532	
9779	065416	020104	043101	042524	
9780	065424	020122	052101	047124	
9781	065432	041040	052111	053440	
9782	065440	044522	052124	047105	
9783	065446	000			
9784					
9785	065447	104	044522	042526	EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
9786	065454	044440	020116	042516	
9787	065462	052125	040522	020114	
9788	065470	043101	042524	020122	
9789	065476	052101	042524	052116	
9790	065504	047511	020116	044502	
9791	065512	020124	051127	052111	
9792	065520	042524	000116		
9793					
9794	065524	051127	052111	020105	EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
9795	065532	052101	042524	052116	
9796	065540	047511	020116	044502	
9797	065546	020124	044504	020104	
9798	065554	047516	020124	042523	
9799	065562	020124	047520	052122	
9800	065570	051040	050505	042525	
9801	065576	052123	000		
9802					
9803	065601	103	047117	051124	EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
9804	065606	046117	042514	020122	
9805	065614	042523	042514	052103	
9806	065622	051440	044527	041524	
9807	065630	020110	047117	042040	
9808	065636	044522	042526	047040	
9809	065644	052117	044440	020116	
9810	065652	040447	041057	000047	
9811					
9812	065660	040503	023516	020124	EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
9813	065666	041501	042503	051523	
9814	065674	042040	044522	042526	
9815	065702	052040	051110	052517	
9816	065710	044107	042440	052111	
9817	065716	042510	020122	047520	
9818	065724	052122	000		
9819					
9820	065727	101	052124	020116	EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/

J14

9821	065734	044502	020124	047506	
9822	065742	020122	042523	055111	
9823	065750	047111	020107	047520	
9824	065756	052122	047040	052117	
9825	065764	041440	042514	051101	
9826	065772	042105	041040	020131	
9827	066000	040515	051523	052502	
9828	066006	020123	047111	052111	
9829	066014	000			
9830					
9831	066015	101	052124	020116	EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
9832	066022	044502	020124	047506	
9833	066030	020122	050117	047520	
9834	066036	044523	042524	050040	
9835	066044	051117	020124	046103	
9836	066052	040505	042522	020104	
9837	066060	054502	046440	051501	
9838	066066	041123	051525	044440	
9839	066074	044516	000124		
9840					
9841	066100	052101	047124	041040	EM51: .ASCIZ /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
9842	066106	052111	041440	042514	
9843	066114	051101	042105	041040	
9844	066122	020131	040515	051523	
9845	066130	052502	020123	047111	
9846	066136	052111	020054	051104	
9847	066144	053111	020105	047111	
9848	066152	047040	052505	051124	
9849	066160	046101	000		
9850					
9851	066163	124	042510	040440	EM52: .ASCIZ /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
9852	066170	052124	020116	044502	
9853	066176	020124	051511	051440	
9854	066204	052105	040440	052106	
9855	066212	051105	052040	046511	
9856	066220	047505	052125	053440	
9857	066226	052111	020110	047516	
9858	066234	051040	050505	042525	
9859	066242	052123	023040	023440	
9860	066250	051105	023522	051440	
9861	066256	052105	000		
9862					
9863	066261	103	047101	052047	EM53: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
9864	066266	051040	040505	020104	
9865	066274	044124	020105	052101	
9866	066302	047124	041040	052111	
9867	066310	043040	047522	020115	
9868	066316	044124	020105	047447	
9869	066324	050120	051517	052111	
9870	066332	023505	050040	051117	
9871	066340	000124			
9872					
9873	066342	042522	042514	051501	EM54: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
9874	066350	020105	047503	046515	
9875	066356	047101	020104	042522	
9876	066364	047503	047107	055111	

K14

9877	066372	042105	053440	042510	
9878	066400	020116	051511	052523	
9879	066406	042105	041040	020131	
9880	066414	047516	026516	042523	
9881	066422	055111	047111	020107	
9882	066430	047520	052122	000	
9883					
9884	066435	124	046511	047505	EM55: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
9885	066442	052125	047440	042516	
9886	066450	051455	047510	020124	
9887	066456	051511	046040	051505	
9888	066464	020123	044124	047101	
9889	066472	032440	030060	046440	
9890	066500	000123			
9891					
9892	066502	044122	030461	042040	EM56: .ASCIZ /RH11 DIDN'T RESPOND TO ADDRESSING/
9893	066510	042111	023516	020124	
9894	066516	042522	050123	047117	
9895	066524	020104	047524	040440	
9896	066532	042104	042522	051523	
9897	066540	047111	000107		
9898					
9899					
9900					
9901					
9902					
9903	066544	042524	052123	021440	DH1: .ASCIZ /TEST # ERR PC PORT # REG ADR CONTENTS/
9904	066552	020040	051105	020122	
9905	066560	041520	020040	047520	
9906	066566	052122	021440	020040	
9907	066574	042522	020107	042101	
9908	066602	020122	047503	052116	
9909	066610	047105	051524	000	
9910	066615	124	051505	020124	DH3: .ASCIZ /TEST # ERR PC REG ADR PORT A PORT B/
9911	066622	020043	042440	051122	
9912	066630	050040	020103	051040	
9913	066636	043505	040440	051104	
9914	066644	050040	051117	020124	
9915	066652	020101	050040	051117	
9916	066660	020124	000102		
9917	066664	020040	020040	020040	DH4: .ASCII / SEIZE ERROR/<CR><LF>
9918	066672	020040	020040	020040	
9919	066700	020040	020040	042523	
9920	066706	055111	020105	020040	
9921	066714	051105	047522	006522	
9922	066722	012			
9923	066723	124	051505	020124	.ASCIZ /TEST # ERR PC PORT # PORT # REG ADR GOOD BAD/
9924	066730	020043	042440	051122	
9925	066736	050040	020103	050040	
9926	066744	051117	020124	020043	
9927	066752	050040	051117	020124	
9928	066760	020043	051040	043505	
9929	066766	040440	051104	043440	
9930	066774	047517	020104	020040	
9931	067002	041040	042101	000	
9932	067007	124	051505	020124	DH5: .ASCIZ /TEST # ERR PC PORT # REG ADR GOOD BAD/

9999	067475	040	020040	020040	DH23:	.ASCII /			SEIZE/<CR><LF>
9990	067502	020040	020040	020040					
9931	067510	020040	020040	051440					
9992	067516	044505	042532	005015					
9993	067524	042524	052123	021440		.ASCIZ /TEST #	ERR PC	PORT #	REG ADR CONTENTS/
9994	067532	020040	051105	020122					
9995	067540	041520	020040	047520					
9996	067546	052122	021440	020040					
9997	067554	042522	020107	042101					
9998	067562	020122	047503	052116					
9999	067570	047105	051524	000					
10000	067575	040	020040	020040	DH26:	.ASCII /			RELSNG/<CR><LF>
10001	067602	020040	020040	020040					
10002	067610	020040	020040	051040					
10003	067616	046105	047123	006507					
10004	067624	012							
10005	067625	124	051505	020124		.ASCIZ /TEST #	ERR PC	PORT #/	
10006	067632	020043	042440	051122					
10007	067640	050040	020103	050040					
10008	067646	051117	020124	000043					
10009	067654	020040	020040	020040	DH31:	.ASCII /			RELSNG RQSTNG/<CR><LF>
10010	067662	020040	020040	020040					
10011	067670	020040	020040	042522					
10012	067676	051514	043516	020040					
10013	067704	050522	052123	043516					
10014	067712	005015							
10015	067714	042524	052123	021440		.ASCIZ /TEST #	ERR PC	PORT #	PORT #/
10016	067722	020040	051105	020122					
10017	067730	041520	020040	047520					
10018	067736	052122	021440	020040					
10019	067744	047520	052122	021440					
10020	067752	000							
10021	067753	124	051505	020124	DH36:	.ASCIZ /TEST #	ERR PC	PORT #/	
10022	067760	020043	042440	051122					
10023	067766	050040	020103	050040					
10024	067774	051117	020124	000043					
10025	070002	042524	052123	021440	DH42:	.ASCIZ /TEST #	ERR PC/		
10026	070010	020040	051105	020122					
10027	070016	041520	000						
10028	070021	040	020040	020040	DH44:	.ASCII /			RELSNG ERROR/<CR><LF>
10029	070026	020040	020040	020040					
10030	070034	020040	020040	051040					
10031	070042	046105	047123	020107					
10032	070050	042440	051122	051117					
10033	070056	005015							
10034	070060	042524	052123	021440		.ASCIZ /TEST #	ERR PC	PORT #	PORT #/
10035	070066	020040	051105	020122					
10036	070074	041520	020040	047520					
10037	070102	052122	021440	020040					
10038	070110	047520	052122	021440					
10039	070116	000							
10040	070117	040	020040	020040	DH46:	.ASCII /			PORT A PORT B/<CR><LF>
10041	070124	020040	020040	020040					
10042	070132	020040	020040	050040					
10043	070140	051117	020124	020101					
10044	070146	050040	051117	020124					

10045	070154	006502	012							
10046	070157	124	051505	020124		.ASCIZ	/TEST	*	ERR PC	RPDS1 RPDS1/
10047	070164	020043	042440	051122						
10048	070172	050040	020103	051040						
10049	070200	042120	030523	020040						
10050	070206	051040	042120	030523						
10051	070214	000								
10052	070215	124	051505	020124	DH55:	.ASCIZ	/TEST	*	ERR PC	PCRT * TIMEOUT VALUE (IN MS)/
10053	070222	020043	042440	051122						
10054	070230	050040	020103	050040						
10055	070236	051117	020124	020043						
10056	070244	052040	046511	047505						
10057	070252	052125	053040	046101						
10058	070260	042525	024040	047111						
10059	070266	046440	024523	000						
10060	070273	044	050122	042101	DH56:	.ASCIZ	/SRPADR/			
10061	070300	000122								
10062										
10063										
10064										.EVEN
10065	070302	001242	001116	001234	DT1:	.WORD	TSTNUM,	\$ERRPC,	PTNBR,	\$BDADR,\$BDDAT,0
10066	070310	001122	001126	000000						
10067	070316	001242	001116	001122	DT3:	.WORD	TSTNUM,	\$ERRPC,\$BDADR,	\$GDDAT,\$BDDAT,0	
10068	070324	001124	001126	000000						
10069	070332	001242	001116	001234	DT5:	.WORD	TSTNUM,	\$ERRPC,PTNBR,\$BDADR,	\$GDDAT,\$BDDAT,0	
10070	070340	001122	001124	001126						
10071	070346	000000								
10072	070350	001242	001116	001240	DT6:	.WORD	TSTNUM,	\$ERRPC,OPPRT,\$BDADR,	\$BDDAT,0	
10073	070356	001122	001126	000000						
10074	070364	001242	001116	001236	DT7:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,PTNBR,\$BDADR,	\$GDDAT,\$BDDAT,0	
10075	070372	001234	001122	001124						
10076	070400	001126	000000							
10077	070404	001242	001116	001236	DT13:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,PTNBR,\$BDADR,	\$BDDAT,0	
10078	070412	001234	001122	001126						
10079	070420	000000								
10080	070422	001242	001116	001236	DT22:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,PTNBR,0		
10081	070430	001234	000000							
10082	070434	001242	001116	001236	DT23:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,\$BDADR,\$BDDAT,0		
10083	070442	001122	001126	000000						
10084	070450	001242	001116	001236	DT31:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,OPPRT,0		
10085	070456	001240	000000							
10086	070462	001242	001116	001236	DT36:	.WORD	TSTNUM,	\$ERRPC,SEIZPT,0		
10087	070470	000000								
10088	070472	001242	001116	001234	DT37:	.WORD	TSTNUM,	\$ERRPC,PTNBR,0		
10089	070500	000000								
10090	070502	001242	001116	000000	DT42:	.WORD	TSTNUM,\$ERRPC,0			
10091	070510	001242	001116	001170	DT46:	.WORD	TSTNUM,\$ERRPC,\$TMP2,\$TMP3,0			
10092	070516	001172	000000							
10093	070522	001242	001116	001240	DT54:	.WORD	TSTNUM,\$ERRPC,OPPRT,SEIZPT,0			
10094	070530	001236	000000							
10095	070534	001242	001116	001236	DT55:	.WORD	TSTNUM,\$ERRPC,SEIZPT,TIME,0			
10096	070542	001262	000000							
10097	070546	001300	000000		DT56:	.WORD	\$RPADR,0			
10098										
10099	070552	000	000	000	DF1:	.BYTE	0,0,0,0,0			
10100	070555	000	000	000						

10101	070557	000	000	000	DF5:	.BYTE	0,0,0,0,0,0
10102	070562	000	000	000			
10103	070565	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0
10104	070570	000	000	000			
10105	070573	000	000	000			
10106	070574	000	000	000	DF31:	.BYTE	0,0,0,0
10107	070577	000	000	000			
10108	070600	000	000	000	DF36:	.BYTE	0,0,0
10109	070603	000	000	000	CF42:	.BYTE	0,0
10110	070605	000	000	000	DF55:	.BYTE	0,0,0,1
10111	070610	001	000	000			
10112	070611	000	000	000	CF56:	.BYTE	0

.EVEN

.SBTTL CONSTANTS, TABLES, ETC

:TABLE OF TEST STARTING ADDRESSES

10125	070612	003070	TSTADR: .WORD	TST1	:STARTING ADDRESS OF TEST	1
10126	070614	004470	.WORD	TST2	:STARTING ADDRESS OF TEST	2
10127	070616	006070	.WORD	TST3	:STARTING ADDRESS OF TEST	3
10128	070620	007470	.WORD	TST4	:STARTING ADDRESS OF TEST	4
10129	070622	010626	.WORD	TST5	:STARTING ADDRESS OF TEST	5
10130	070624	011764	.WORD	TST6	:STARTING ADDRESS OF TEST	6
10131	070626	012754	.WORD	TST7	:STARTING ADDRESS OF TEST	7
10132	070630	013744	.WORD	TST10	:STARTING ADDRESS OF TEST	10
10133	070632	014366	.WORD	TST11	:STARTING ADDRESS OF TEST	11
10134	070634	015010	.WORD	TST12	:STARTING ADDRESS OF TEST	12
10135	070636	016366	.WORD	TST13	:STARTING ADDRESS OF TEST	13
10136	070640	017744	.WORD	TST14	:STARTING ADDRESS OF TEST	14
10137	070642	021246	.WORD	TST15	:STARTING ADDRESS OF TEST	15
10138	070644	022550	.WORD	TST16	:STARTING ADDRESS OF TEST	16
10139	070646	024010	.WORD	TST17	:STARTING ADDRESS OF TEST	17
10140	070650	024676	.WORD	TST20	:STARTING ADDRESS OF TEST	20
10141	070652	025564	.WORD	TST21	:STARTING ADDRESS OF TEST	21
10142	070654	026744	.WORD	TST22	:STARTING ADDRESS OF TEST	22
10143	070656	030124	.WORD	TST23	:STARTING ADDRESS OF TEST	23
10144	070660	031134	.WORD	TST24	:STARTING ADDRESS OF TEST	24
10145	070662	032144	.WORD	TST25	:STARTING ADDRESS OF TEST	25
10146	070664	033352	.WORD	TST26	:STARTING ADDRESS OF TEST	26
10147	070666	034560	.WORD	TST27	:STARTING ADDRESS OF TEST	27
10148	070670	036554	.WORD	TST30	:STARTING ADDRESS OF TEST	30
10149	070672	037250	.WORD	TST31	:STARTING ADDRESS OF TEST	31
10150	070674	040510	.WORD	TST32	:STARTING ADDRESS OF TEST	32
10151	070676	041750	.WORD	TST33	:STARTING ADDRESS OF TEST	33
10152	070700	043040	.WORD	TST34	:STARTING ADDRESS OF TEST	34
10153	070702	044130	.WORD	TST35	:STARTING ADDRESS OF TEST	35
10154	070704	044744	.WORD	TST36	:STARTING ADDRESS OF TEST	36
10155	070706	045560	.WORD	TST37	:STARTING ADDRESS OF TEST	37
10156	070710	046542	.WORD	TST40	:STARTING ADDRESS OF TEST	40

10157	070712	047524	.WORD	TST41	; STARTING ADDRESS OF TEST 41
10158	070714	051132	.WORD	TST42	; STARTING ADDRESS OF TEST 42
10159	070716	052540	.WORD	TST43	; STARTING ADDRESS OF TEST 43
10160	070720	053422	.WORD	TST44	; STARTING ADDRESS OF TEST 44
10161	070722	054310	.WORD	TST45	; STARTING ADDRESS OF TEST 45
10162	070724	055256	.WORD	TST46	; STARTING ADDRESS OF TEST 45

;ATTENTION BIT TABLE

10163					
10164					
10165					
10166	070726	001	ATABIT: .BYTE	1	; ATTENTION BIT FOR DRIVE 0
10167	070727	002	.BYTE	2	; ATTENTION BIT FOR DRIVE 1
10168	070730	004	.BYTE	4	; ATTENTION BIT FOR DRIVE 2
10169	070731	010	.BYTE	10	; ATTENTION BIT FOR DRIVE 3
10170	070732	020	.BYTE	20	; ATTENTION BIT FOR DRIVE 4
10171	070733	040	.BYTE	40	; ATTENTION BIT FOR DRIVE 5
10172	070734	100	.BYTE	100	; ATTENTION BIT FOR DRIVE 6
10173	070735	200	.BYTE	200	; ATTENTION BIT FOR DRIVE 7

10174					
10175	070736	000046	MAXTN: .WORD	\$TN-1	; MAXIMUM TEST NUMBER
10176					
10177		000001	.END		

G15

DVA	=	004000	1258#																
ECH	=	000100	1287#																
ECI	=	004000	1401#																
EMTVEC	=	000030	1192#	1951*	1952*														
EM1		062506	1607	9485#															
EM10		063113	1656	9540#															
EM11		063143	1663	9546#															
EM12		063227	1670	9556#															
EM13		063277	1677	9564#															
EM14		063364	1684	9574#															
EM15		063427	1691	9581#															
EM16		063506	1698	9590#															
EM17		063561	1705	9599#															
EM2		062527	1614	9489#															
EM20		063640	1712	9608#															
EM21		063720	1719	9617#															
EM22		063773	1726	9626#															
EM23		064060	1733	9636#															
EM24		064126	1740	9644#															
EM25		064205	1748	9653#															
EM26		064250	1756	9660#															
EM27		064335	1763	9670#															
EM3		062551	1621	9494#															
EM30		064413	1770	9679#															
EM31		064510	1777	9691#															
EM32		064565	1784	9700#															
EM33		064636	1791	9708#															
EM34		064740	1798	9720#															
EM35		065043	1805	9733#															
EM36		065122	1812	9742#															
EM37		065174	1819	9750#															
EM4		062633	1628	9504#															
EM40		065242	1826	9758#															
EM41		065317	1833	9767#															
EM42		065361	1840	9774#															
EM43		065447	1847	9785#															
EM44		065524	1854	9794#															
EM45		065601	1861	9803#															
EM46		065660	1868	9812#															
EM47		065727	1875	9820#															
EM5		062664	1635	9510#															
EM50		066015	1882	9831#															
EM51		066100	1889	9841#															
EM52		066163	1896	9851#															
EM53		066261	1903	9863#															
EM54		066342	1910	9873#															
EM55		066435	1917	9884#															
EM56		066502	1924	9892#															
EM6		062732	1642	9518#															
EM7		063032	1649	9530#															
ENTERA		062117	1991	9432#															
ERR	=	040000	1276#	6311	6513														
ERRVEC	=	000004	1185#	1963	1964*	1975*	8610*	8611*	8620*	8629*	8681	8682*	8684*	8687*					
EXEC		002602	2027	2034#	2115	2313	2550	2788	2967	3143	3288	3431	3511	3602	3806				
			4011	4212	4411	4589	4711	4841	5021	5199	5348	5501	5690	5876	6155				
			6252	6454	6663	6832	6998	7135	7275	7423	7581	7795	7998	8112	8242				

MSTCK = 000010	1303#												
MWR = 000040	1305#												
MXF = 001000	1232#												
NBA = 100000	1336#												
NED = 010000	1235#	2135	2139	2156	2160								
NEM = 004000	1234#												
NHS = 002000	1369#	1387#											
NORTA = 000001	1931#	2444	2450	2681	2687	2934	2940	3113	3119	3208	3214	3353	3359
	3485	3491	3565	3571	3770	3776	3974	3980	4156	4162	4357	4363	4391#
	4516	4521	4684	4689	4806	4811	4876#	4943#	4987	4992	5056#	5123#	5167
	5172	5317	5322	5466	5471	5656	5661	5845	5850	6021	6026	6127	6132
	6222	6227	6358#	6395	6401	6406#	6560#	6597	6603	6608#	6797	6802	6966
	6971	7095	7100	7232	7237	7392	7397	7540	7545	7698	7703	7912	7917
	8038#	8082	8088	8093#	8152#	8196	8202	8207#	8370	8375	8535	8540	
	2028	9449#											
NOCLOC 062245	1566#												
NOSEIZ 001246	2073	9473#											
NTRH11 062454	1422#	1432#											
OCYL = 100000	1399#												
OFREV = 000200	1395#												
OF100 = 000004	1396#												
OF200 = 000010	1393#												
OF25 = 000001	1397#												
OF400 = 000020	1394#												
OF50 = 000002	1398#												
OF800 = 000040	1430#												
OPE = 020000	1294#												
OPI = 020000	1563#	2340*	2577*	3164*	3309*	3619*	3823*	4085*	4286*	4617*	4739*	4867*	5047*
OPPRT 001240	5227*	5376*	5529*	5718*	5948*	6054*	6277*	6479*	6688*	6857*	7024*	7161*	7605*
	7819*	8268*	8433*	10072	10084	10093							
	1230#												
GR = 000200	1284#												
PAR = 000010	1227#												
PAT = 000020	1233#												
PGE = 002000	1271#	2216	2218	2222	2248	2250	2254	2353	2415	2590	2652	2831	2871
PGM = 001000	2905	3010	3050	3084	3179	3324	3456	3536	3632	3673	3716	3739	3836
	3877	3920	3943	4125	4326	4630	4653	4752	4775	4889	4956	5069	5136
	5240	5286	5389	5435	5542	5594	5625	5731	5783	5814	5936	5959	5990
	6042	6065	6096	6191	6766	6935	7064	7201	7361	7509	7636	7667	7850
	7881	8051	8165	8339	8504								
	1275#												
PIP = 020000	1102#												
PIRQ = 177772	1196#												
PIRQVE = 000240	1372#	1389#											
PLU = 020000	1557#	1993#	1994	1998	2001	2005	2017	2020	2126	2127	2172	2173	2210
PORTA 001224	2211	2278	2335	2336	2350	2351	2416	2435	2437	2443	2575	2576	2577
	2603	2604	2653	2672	2674	2680	2697	2795	2796	2803	2825	2826	2906
	2925	2927	2933	2992	2993	3061	3062	3085	3104	3106	3112	3161	3182
	3170	3171	3180	3199	3201	3207	3222	3223	3309	3325	3344	3346	3352
	3384	3385	3408	3409	3438	3439	3440	3457	3476	3478	3484	3537	3556
	3558	3564	3614	3615	3629	3630	3667	3668	3710	3711	3730	3731	3740
	3759	3761	3769	3821	3822	3823	3854	3855	3897	3898	3944	3953	3965
	3973	4022	4030	4037	4038	4082	4083	4091	4116	4117	4126	4145	4147
	4155	4223	4231	4258	4259	4286	4327	4346	4348	4356	4372	4373	4422
	4430	4437	4438	4486	4505	4507	4515	4536	4537	4599	4612	4613	4627
	4628	4644	4645	4654	4673	4675	4683	4721	4737	4738	4739	4776	4795

J15

4797	4805	4851	4867	4868	4869	4892	4893	4903	4925	4926	4947	4948
4957	4976	4978	4986	5031	5044	5045	5062	5063	5075	5076	5088	5089
5137	5156	5158	5166	5209	5225	5226	5227	5249	5250	5287	5306	5308
5316	5358	5371	5372	5386	5387	5426	5427	5436	5455	5457	5465	5511
5527	5528	5529	5555	5556	5597	5598	5608	5616	5617	5626	5645	5647
5555	5700	5713	5714	5728	5729	5776	5777	5789	5790	5815	5834	5836
5844	5886	5899	5905	5906	5948	5962	5963	5973	5981	5982	5991	6010
6012	6020	6036	6037	6053	6058	6059	6071	6072	6097	6116	6118	6126
6165	6192	6211	6213	6221	6262	6274	6275	6301	6302	6365	6384	6386
6394	6430	6464	6479	6491	6492	6567	6586	6588	6596	6612	6613	6673
6685	6686	6714	6715	6735	6736	6757	6758	6767	6786	6788	6796	6842
6857	6892	6893	6936	6955	6957	6965	7008	7021	7022	7065	7084	7086
7094	7145	7161	7185	7186	7202	7221	7223	7231	7285	7293	7294	7295
7352	7353	7362	7381	7383	7391	7433	7474	7475	7510	7529	7531	7539
7591	7602	7603	7612	7613	7623	7624	7629	7630	7642	7643	7668	7687
7689	7697	7805	7819	7823	7824	7832	7833	7853	7854	7864	7872	7873
7882	7901	7903	7911	8008	8016	8052	8071	8073	8081	8122	8133	8142
8143	8144	8156	8157	8166	8185	8187	8195	8252	8268	8269	8270	8311
8330	8331	8340	8359	8361	8369	8417	8430	8431	8441	8442	8505	8524
8526	8534											
2004	9441#											
1558#	1998*	1999*	2000*	2003*	2011	2147	2148	2189	2190	2242	2243	2280
2338	2339	2340	2366	2367	2420	2431	2432	2449	2460	2572	2573	2587
2588	2657	2668	2669	2686	2813	2814	2882	2883	2910	2921	2922	2939
2974	2975	2982	3004	3005	3089	3100	3101	3118	3164	3184	3195	3196
3213	3239	3240	3263	3264	3306	3307	3315	3316	3329	3340	3341	3358
3367	3368	3461	3472	3473	3490	3518	3519	3520	3541	3552	3553	3570
3617	3618	3619	3650	3651	3693	3694	3744	3755	3756	3775	3818	3819
3833	3834	3871	3872	3914	3915	3934	3935	3948	3959	3960	3979	4025
4057	4058	4085	4130	4141	4142	4161	4171	4172	4226	4238	4239	4283
4284	4292	4317	4318	4331	4342	4343	4362	4425	4457	4458	4490	4501
4502	4520	4553	4554	4603	4615	4616	4617	4658	4669	4670	4688	4725
4734	4735	4749	4750	4766	4767	4780	4791	4792	4810	4855	4864	4865
4882	4883	4895	4896	4908	4909	4961	4972	4973	4991	5035	5047	5048
5049	5072	5073	5083	5105	5106	5127	5128	5141	5152	5153	5171	5213
5222	5223	5237	5238	5277	5278	5291	5302	5303	5321	5362	5374	5375
5376	5398	5399	5440	5451	5452	5470	5515	5524	5525	5539	5540	5587
5588	5600	5601	5630	5641	5642	5660	5704	5716	5717	5718	5744	5745
5786	5787	5797	5805	5806	5819	5830	5831	5849	5890	5913	5914	5920
5921	5947	5952	5953	5965	5966	5995	6006	6007	6025	6054	6068	6069
6079	6087	6088	6101	6112	6113	6131	6169	6196	6207	6208	6226	6266
6277	6289	6290	6369	6380	6381	6400	6410	6411	6468	6476	6477	6503
6504	6571	6582	6583	6602	6632	6677	6688	6723	6724	6771	6782	6783
6801	6846	6854	6855	6883	6884	6904	6905	6926	6927	6940	6951	6952
6970	7012	7024	7048	7049	7069	7080	7081	7099	7149	7158	7159	7206
7217	7218	7236	7289	7326	7327	7366	7377	7378	7396	7437	7441	7442
7443	7500	7501	7514	7525	7526	7544	7595	7605	7609	7610	7618	7619
7639	7640	7650	7658	7659	7672	7683	7684	7702	7809	7816	7817	7826
7827	7837	7838	7843	7844	7856	7857	7886	7897	7898	7916	8012	8019
8028	8029	8030	8042	8043	8056	8067	8068	8087	8126	8130	8170	8181
8182	8201	8256	8265	8266	8276	8277	8344	8355	8356	8374	8421	8433
8434	8435	8476	8495	8496	8509	8520	8521	8539				
2010	9445#											
PORTBI 062217	1559#	2018*	2019*	5898	6173							
PORTC 001230	1418#											
PRE = 000020	1119#											
PRO = 000000												

PORTA 062171
 PORTB 001226

PORTBI 062217
 PORTC 001230
 PRE = 000020
 PRO = 000000

TST10 013744 3427# 10132
TST11 014366 3507# 10133
TST12 015010 3598# 10134
TST13 016366 3802# 10135
TST14 017744 4007# 10136
TST15 021246 4208# 10137
TST16 022550 4407# 10138
TST17 024010 4585# 10139
TST2 004470 2309# 10126
TST20 024676 4707# 10140
TST21 025564 4837# 10141
TST22 026744 5017# 10142
TST23 030124 5195# 10143
TST24 031134 5344# 10144
TST25 032144 5497# 10145
TST26 033352 5686# 10146
TST27 034560 5872# 10147
TST3 006070 2546# 10127
TST30 036554 6151# 10148
TST31 037250 6248# 10149
TST32 040510 6457# 10150
TST33 041750 6659# 10151
TST34 043040 6828# 10152
TST35 044130 6994# 10153
TST36 044744 7131# 10154
TST37 045560 7271# 10155
TST4 007470 2784# 10128
TST40 046542 7419# 10156
TST41 047524 7577# 10157
TST41B 050600 7611 7625
TST42 051132 7708 7791#
TST42B 052206 7825 7839
TST43 052540 7922 7994#
TST44 052422 8108# 10160
TST45 054310 8238# 10161
TST46 055256 8403# 10162
TST5 010626 2963# 10129
TST6 011764 3139# 10130
TST7 012754 3284# 10131
TJF = 000100 1365# 1383#
TYPDS = 104405 8577 8584 8804 9402#
TYPE = 104401 1979 1991 1996 2004 2010 2016 2028 2036 2047 2054 2069 2072 2073
8571 8578 8585 8740 8748 8768 8785 8787 8790 8792 88J7 8812 8850
8946 9021 9073 9127 9128 9131 9144 9155 9174 9222 9228 9233 9237
9242 9243 9245 9248 9252 9317 9319 9398#
2071 8776 8800 9130 9399#
TYPQC = 104402 2071 8776 8800 9130 9399#
TYPON = 104404 9401#
TYPOS = 104403 2007 2013 9400#
UNS = 040000 1295#
UPE = 020000 1236#
US1 = 000001 1223#
US2 = 000002 1224#
US4 = 000004 1225#
UMR = 000010 1417#
VUF = 000002 1416#
VL30 = 010000 1371#

7712#
10158
7926#
10159

VV = 000100

1268*	2419	2423	2444	2450	2656	2660	2681	2687	2871	2909	2913	2924
2940	3050	3084	3098	3092	3179	3183	3187	3324	3328	3332	3456	3460
3464	3536	3540	3544	3632	3673	3716	3739	3743	3747	3836	3877	3920
3943	3947	3951	4125	4129	4133	4326	4320	4334	4489	4493	4630	4653
4657	4661	4752	4775	4779	4783	4889	4956	4960	4964	5069	5136	5140
5144	5240	5286	5290	5294	5389	5435	5439	5443	5542	5594	5625	5629
5633	5731	5783	5814	5818	5822	5936	5959	5990	5994	5998	6042	6065
6096	6100	6104	6191	6195	6199	6368	6372	6570	6574	6766	6770	6774
6935	6939	6943	7064	7068	7072	7201	7205	7209	7361	7365	7369	7509
7513	7517	7636	7667	7671	7675	7850	7881	7885	7889	8051	8055	8059

VVSET = 000001

8165	8169	8173	8339	8343	8347	8504	8508	8512	2681	2687	2905	2931
1931*	2353	2415	2441	2444	2450	2590	2652	2678	2681	2687	2905	2931
2934	2940	3042*	3084	3110	3113	3114	3119	3120	3179	3205	3208	3209
3214	3215	3324	3350	3353	3354	3359	3360	3456	3482	3485	3486	3491
3492	3536	3562	3565	3566	3571	3572	3632	3739	3765	3767	3770	3771
3776	3777	3836	3943	3969	3971	3974	3975	3980	3981	4125	4151	4153
4156	4157	4162	4163	4326	4352	4354	4357	4358	4363	4364	4511	4513
4516	4521	4630	4653	4679	4681	4684	4689	4752	4775	4801	4803	4806
4811	4889	4956	4982	4984	4987	4992	5069	5136	5162	5164	5167	5172
5240	5286	5312	5314	5317	5322	5389	5435	5461	5463	5466	5471	5542
5594	5625	5651	5653	5656	5651	5731	5783	5814	5840	5842	5845	5950
5959	5990	6016	6018	6021	6026	6065	6096	6122	6124	6127	6132	6191
6217	6222	6227	6390	6392	6395	6396	6401	6402	6592	6594	6597	6598
6603	6604	6766	6792	6794	6797	6802	6935	6961	6963	6966	6971	7064
7090	7095	7100	7201	7227	7232	7237	7361	7387	7389	7392	7397	7509
7535	7537	7540	7545	7636	7667	7693	7695	7698	7703	7850	7881	7907
7909	7912	7917	8051	8077	8079	8082	8083	8088	8089	8165	8191	8193
8196	8197	8202	8203	8339	8365	8367	8370	8375	8504	8530	8532	8535

WAO = 000002
WATCH 001254

1427*	1569*	2329*	2387	2566*	2624	2802*	2890	2981*	3069	3155*	3267	3300*	3412
6288*	6297	6490*	6499	7034*	7035	7042*	7052	7171*	7172	7179*	7189	8287*	
8305	8452*	8470	8635	8637*	8639*								

WCE = 040000
WCF = 000040
WCU = 000001
WLE = 004000
WRL = 004000
WRU = 000400
WSU = 000004
SAUTOB 001134
SBOARD 001122

1237*	1286*	1359*	1377*	1385*	1361*	1379*	1522*	1517*	2131*	2132*	2152*	2153*	2176*	2177*	2193*	2194*	2214*	2215*	2229*	2230*																																																																																																																							
2246*	2247*	2261*	2262*	2262*	2342*	2343*	2413*	2414*	2461*	2462*	2466*	2467*	2471*	2482*	2483*	2487*	2491*	2492*	2496*	2497*	2501*	2502*	2598*	2698*	2729*	2729*	2729*	2729*	2754*	2758*	2759*	2817*	2817*	3008*	3009*	3023*	3024*	3048*	3049*	3082*	3083*	3177*	3178*	3226*	3227*	3243*	3244*	3322*	3323*	3371*	3372*	3388*	3389*	3454*	3455*	3534*	3535*	3621*	3622*	3654*	3655*	3671*	3672*	3697*	3698*	3714*	3715*	3737*	3738*	3825*	3826*	3858*	3859*	3875*	3876*	3901*	3902*	3918*	3919*	3941*	3942*	4041*	4042*	4061*	4062*	4098*	4099*	4123*	4124*	4175*	4176*	4242*	4243*	4262*	4263*	4299*	4300*	4324*	4325*	4376*	4377*	4441*	4442*	4461*	4462*	4483*	4484*	4540*	4541*	4557*	4558*	4619*	4620*	4651*	4652*	4741*	4742*	4773*	4774*	4890*	4891*	4912*	4913*	4929*	4930*	4954*	4955*	5070*	5071*	5092*	5093*	5109*	5110*	5134*	5135*	5229*	5230*	5257*	5258*

E16

5284*	5285*	5378*	5379*	5406*	5407*	5433*	5434*	5531*	5532*	5564*	5565*	5572*
5573*	5595*	5596*	5623*	5624*	5720*	5721*	5753*	5754*	5761*	5762*	5784*	5785*
5812*	5813*	5934*	5935*	5960*	5961*	5988*	5989*	6040*	6041*	6066*	6067*	6094*
6095*	6189*	6190*	6309*	6310*	6328*	6329*	6342*	6343*	6362*	6363*	6414*	6415*
6511*	6512*	6530*	6531*	6544*	6545*	6564*	6565*	6616*	6617*	6697*	6698*	6727*
6728*	6739*	6740*	6764*	6765*	6866*	6867*	6896*	6897*	6908*	6909*	6933*	6934*
7062*	7063*	7199*	7200*	7313*	7314*	7334*	7335*	7359*	7360*	7461*	7462*	7482*
7483*	7507*	7508*	7637*	7638*	7665*	7666*	7715*	7716*	7723*	7724*	7732*	7733*
7740*	7741*	7749*	7750*	7757*	7758*	7851*	7852*	7879*	7880*	7929*	7930*	7937*
7938*	7946*	7947*	7954*	7955*	7963*	7964*	7971*	7972*	8022*	8023*	8049*	8050*
8136*	8137*	8163*	8164*	8294*	8295*	8314*	8315*	8337*	8338*	8459*	8460*	8479*
8480*	8502*	8503*	10065	10067	10069	10072	10074	10077	10082			
1519*	2130*	2134	2138	2151*	2155	2159	2175*	2179	2183	2192*	2196	2200
2213*	2217	2221	2228*	2232	2236	2245*	2249	2253	2260*	2264	2268	2281*
2282	2341*	2345	2352*	2356	2430*	2436*	2442*	2448*	2463*	2468*	2473*	2478*
2483*	2488*	2493*	2498*	2503*	2508*	2513*	2518*	2523*	2578*	2582	2589*	2593
2667*	2673*	2679*	2685*	2700*	2705*	2710*	2715*	2720*	2725*	2730*	2735*	2740*
2745*	2750*	2755*	2760*	2816*	2820	2828*	2832	2836	2843*	2847	2851	2868*
2872	2876	2920*	2926*	2932*	2938*	2995*	2999	3007*	3011	3015	3022*	3026
3030	3047*	3051	3055	3099*	3105*	3111*	3117*	3194*	3200*	3206*	3212*	3225*
3229	3233	3242*	3246	3250	3339*	3345*	3351*	3357*	3370*	3374	3378	3387*
3391	3395	3471*	3477*	3483*	3489*	3551*	3557*	3563*	3569*	3620*	3624	3631*
3635	3653*	3657	3661	3670*	3674	3678	3696*	3700	3704	3713*	3717	3721
3754*	3760*	3768*	3774*	3824*	3828	3835*	3839	3857*	3861	3865	3874*	3878
3882	3900*	3904	3908	3917*	3921	3925	3958*	3964*	3972*	3978*	4040*	4044
4048	4060*	4064	4068	4097*	4101	4105	4140*	4146*	4154*	4160*	4174*	4178
4182	4241*	4245	4249	4261*	4265	4269	4298*	4302	4306	4341*	4347*	4355*
4361*	4375*	4379	4383	4440*	4444	4448	4460*	4464	4468	4500*	4506*	4514*
4519*	4539*	4543	4547	4556*	4560	4564	4618*	4622	4629*	4633	4668*	4674*
4682*	4687*	4740*	4744	4751*	4755	4790*	4796*	4804*	4809*	4897*	4902*	4904
4911*	4915	4919	4928*	4932	4936	4971*	4977*	4985*	4990*	5077*	5082*	5094
5091*	5095	5099	5108*	5112	5116	5151*	5157*	5165*	5170*	5228*	5232	5239*
5243	5256*	5260	5264	5301*	5307*	5315*	5320*	5377*	5381	5388*	5392	5405*
5409	5413	5450*	5456*	5464*	5469*	5530*	5534	5541*	5545	5563*	5567	5571*
5574	5602*	5607*	5609	5640*	5646*	5654*	5659*	5719*	5723	5730*	5734	5752*
5756	5760*	5763	5791*	5796*	5798	5829*	5835*	5843*	5848*	5933*	5937	5941
5967*	5972*	5974	6005*	6011*	6019*	6024*	6039*	6043	6047	6073*	6078*	6080
6111*	6117*	6125*	6130*	6206*	6212*	6220*	6225*	6308*	6312	6316	6327*	6331
6341*	6345	6349	6379*	6385*	6393*	6399*	6413*	6417	6421	6510*	6514	6518
6529*	6533	6543*	6547	6551	6581*	6587*	6595*	6601*	6615*	6619	6623	6696*
6700	6704	6726*	6730	6738*	6742	6781*	6787*	6795*	6800*	6865*	6869	6873
6895*	6899	6907*	6911	6950*	6956*	6964*	6969*	7079*	7085*	7093*	7098*	7216*
7222*	7230*	7235*	7312*	7316	7320	7333*	7337	7341	7376*	7382*	7390*	7395*
7460*	7464	7468	7481*	7485	7489	7524*	7530*	7538*	7543*	7644*	7649*	7651
7682*	7688*	7696*	7701*	7714*	7718	7722*	7725	7731*	7735	7739*	7742	7748*
7752	7756*	7759	7858*	7863*	7865	7896*	7902*	7910*	7915*	7928*	7932	7936*
7939	7945*	7949	7953*	7956	7962*	7966	7970*	7973	8032*	8033	8066*	8072*
8080*	8086*	8146*	8147	8180*	8186*	8194*	8200*	8293*	8297	8313*	8317	8354*
8360*	8368*	8373*	8458*	8462	8478*	8482	8519*	8525*	8533*	8538*	10065	10067
10069	10072	10074	10077	10082								
1545*	8740	8760	9073	9262								
8852*	8862*	8869	8878*	8883*								
9106*	9406											
1505*	1942	1943	1951	1955	1956	1957						
1537*	1538*											
1537*	1538*											

SBODAT 001126

SBELL 001202
 SCHARC 057644
 SCKSWR 060574
 SCMTAG 001100
 SCM1 = 000001
 SCM2 = 000002

STMP1 001166

1539*	2354*	2355*	2357	2422*	2423*	2424	2438	2591*	2592*	2594	2659*	2660*
2661	2675	2912*	2913*	2914	2928	3091*	3092*	3093	3107	3186*	3187*	3188
3202	3331*	3332*	3333	3347	3463*	3464*	3465	3479	3543*	3544*	3545	3559
3633*	3634*	3636	3746*	3747*	3748	3762	3837*	3838*	3840	3950*	3951*	3952
3966	4132*	4133*	4134	4148	4333*	4334*	4335	4349	4492*	4493*	4494	4508
4631*	4632*	4634	4660*	4661*	4662	4676	4753*	4754*	4756	4782*	4783*	4784
4798	4963*	4964*	4965	4979	5143*	5144*	5145	5159	5241*	5242*	5241	5293*
5294*	5295	5309	5390*	5391*	5393	5442*	5443*	5444	5458	5543*	5544*	5546
5632*	5633*	5634	5648	5732*	5733*	5735	5821*	5822*	5823	5837	5997*	5998*
5999	6013	6103*	6104*	6105	6119	6198*	6199*	6200	6214	6371*	6372*	6373
6387	6573*	6574*	6575	6589	6773*	6774*	6775	6789	6942*	6943*	6944	6958
7071*	7072*	7073	7087	7208*	7209*	7210	7224	7368*	7369*	7370	7384	7516*
7517*	7518	7532	7674*	7675*	7676	7690	7888*	7889*	7890	7904	8025*	8026*
8034	8058*	8059*	8060	8074	8139*	8140*	8148	8172*	8173*	8174	8188	8346*

STMP2 001170

1540*	2417*	2418	2430	2442	2444*	2445	2654*	2655	2667	2679	2681*	2682
2907*	2908	2920	2932	2934*	2935	3086*	3087	3099	3111	3113*	3114	3181*
3182	3194	3206	3208*	3209	3326*	3327	3339	3351	3353*	3354	3458*	3459
3471	3483	3485*	3486	3538*	3539	3551	3563	3565*	3566	3741*	3742	3754
3768	3770*	3771	3945*	3946	3958	3972	3974*	3975	4127*	4128	4140	4154
4156*	4157	4328*	4329	4341	4355	4357*	4358	4487*	4488	4500	4514	4516
4655*	4656	4668	4682	4684	4777*	4778	4790	4804	4806	4958*	4959	4971
4985	4987	5138*	5139	5151	5165	5167	5288*	5289	5301	5315	5317	5437*
5438	5450	5464	5466	5627*	5628	5640	5654	5656	5816*	5817	5829	5843
5845	5992*	5993	6005	6019	6021	6098*	6099	6111	6125	6127	6193*	6194
6206	6220	6222	6366*	6367	6379	6393	6395*	6396	6568*	6569	6581	6595
6597*	6598	6768*	6769	6781	6795	6797	6937*	6938	6950	6964	6366	7066*
7067	7079	7093	7095	7203*	7204	7216	7230	7232	7363*	7364	7376	7390
7392	7511*	7512	7524	7538	7540	7669*	7670	7682	7696	7699	7883*	7884
7896	7910	7912	8053*	8054	8066	8080	8082*	8083	8167*	8168	8180	8194
8196*	8197	8341*	8342	8354	8368	8370	8506*	8507	8519	8533	8535	10091

STMP3 001172

1541*	2421*	2422	2426	2448	2450*	2451	2658*	2659	2673	2685	2687*	2688
2911*	2912	2926	2928	2940*	2941	3090*	3091	3105	3117	3119*	3120	3185*
3186	3200	3212	3214*	3215	3330*	3331	3345	3357	3359*	3360	3462*	3463
3477	3489	3491*	3492	3542*	3543	3557	3569	3571*	3572	3745*	3746	3760
3774	3776*	3777	3949*	3950	3964	3978	3980*	3981	4131*	4132	4146	4160
4162*	4163	4332*	4333	4347	4361	4363*	4364	4491*	4492	4506	4519	4521
4659*	4660	4674	4687	4662	4781*	4782	4796	4809	4811	4962*	4963	4977
4990	4992	5142*	5143	5157	5170	5172	5292*	5293	5307	5320	5322	5441*
5442	5452	5469	5471	5631*	5632	5646	5659	5661	5820*	5821	5835	5848
5850	5996*	5997	6011	6024	6026	6102*	6103	6117	6130	6132	6197*	6198
6212	6225	6227	6370*	6371	6385	6399	6401*	6402	6572*	6573	6587	6601
6603*	6604	6772*	6773	6787	6800	6802	6941*	6942	6956	6969	6971	7070*
7071	7085	7098	7100	7207*	7208	7222	7235	7237	7367*	7368	7382	7395
7397	7515*	7516	7530	7543	7545	7673*	7674	7688	7701	7703	7887*	7888
7902	7915	7917	8057*	8058	8072	8086	8088*	8089	9171*	8172	8186	8200

STMP4 001174

8202*	8203	8345*	8346	8360	8373	8375	8510*	8511	8525	8538	8540	10091
1542*	2138*	2139*	2140	2159*	2160*	2161	2183*	2184*	2185	2200*	2201*	2202
2221*	2222*	2223	2236*	2237*	2238	2253*	2254*	2255	2268*	2269*	2270	2836*
2837*	2838	2851*	2852*	2853	2876*	2877*	2878	3015*	3016*	3017	3030*	3031*
3032	3055*	3056*	3057	3233*	3234*	3235	3250*	3251*	3252	3378*	3379*	3380
3395*	3396*	3397	3661*	3662*	3663	3678*	3679*	3680	3704*	3705*	3706	3721*
3722*	3723	3865*	3866*	3867	3882*	3883*	3884	3909*	3909*	3910	3925*	3926*
3927	4048*	4049*	4050	4068*	4069*	4070	4105*	4106*	4107	4182*	4183*	4184
4249*	4250*	4251	4269*	4270*	4271	4306*	4307*	4308	4383*	4384*	4385	4448*
4449*	4450	4468*	4469*	4470	4547*	4548*	4549	4564*	4565*	4566	4919*	4920*

M16

	5356	5123	5179	5194	5218	5252	5271	5328	5343	5367	5401	5420	5479	5496	5520
	5552	5559	5583	5667	5685	5709	5741	5748	5772	5257	5871	5895	5901	5909	5928
	6032	6139	6150	6175	6182	6234	6247	6270	6279	6284	6292	6304	6323	6337	6357
	6407	6436	6449	6472	6481	6486	6494	6506	6525	6539	6559	6609	6639	6658	6681
	6690	6719	6748	6753	6808	6827	6850	6859	6888	6917	6922	6978	6993	7017	7026
	7030	7038	7044	7057	7105	7115	7130	7154	7163	7167	7175	7191	7194	7242	7254
	7270	7297	7302	7308	7329	7348	7403	7418	7445	7450	7456	7477	7496	7552	7576
	7766	7790	7981	7993	8095	8107	8210	8214	8217	8237	8261	8272	8279	8284	8289
	8308	8326	8382	8402	8426	8437	8444	8449	8454	8473	8491	8549	8600	8604	8655
	8659	8663	8721	8762	8818	8888	8965	9032	9101	9116	9179	9203	9271	9324	9369
	9413	9417	9479	9483	10117	10121									
SWITCH	1063#	4885	5065	5590	5779	5955	6061	7632	7846						
SWRSU	1197#	1961#													
TESTAG	1063#	2121	2319	2556	2794	2973	3149	3294	3437	3517	3608	3812	4017	4218	4417
	4595	4717	4847	5027	5205	5354	5507	5696	5882	6161	6258	6460	6669	6838	7004
	7141	7281	7429	7587	7801	8004	8118	8248	8413						
TIMER	1063#	2324	2561	2797	2976	3150	3295	6283	6485	7029	7037	7166	7174		
TRMTRP	9390#														
TYPBIN	1197#														
TYPDEC	1197#	8575	8582	8802											
TYPNAM	1197#														
TYPNUM	1197#														
TYPOCS	1197#	2005	2011												
TYPOCT	1197#	8774	8799	9129											
TYPTXT	1197#	8571	8578												
USEREO	8547#	8558													
\$\$CMRE	1498#	1537													
\$\$CMTM	1498#	1538	1539	1540	1541	1542									
\$\$ESCA	1197#														
\$\$NEWT	1197#	2098	2291	2528	2766	2947	3127	3272	3418	3498	3579	3783	3988	4190	4393
	4573	4695	4818	4998	5179	5328	5478	5667	5857	6139	6234	6436	6639	6808	6978
	7115	7254	7403	7552	7766	7981	8095	8217	8382						
\$\$SET	9390#	9399	9400	9401	9402	9404	9406	9407	9408	9409	9410	9411			
\$\$SKIP	1197#														
.EQUAT	1063#	1087													
.HEADE	1063#	1064													
.SEYUP	1063#	1931													
.SWRHI	1063#	1075													
.SWRLO	1063#	1085#													
.\$ACTI	1063#	1478													
.\$CATC	1063#	1468													
.\$CMTA	1063#	1498													
.\$EOP	1063#	8547													
.\$ERRO	1063#	8719													
.\$ERRT	1063#	8760													
.\$RDOC	1063#	9269													
.\$READ	1063#	9030													
.\$SAVE	1063#	9322													
.\$SCOP	1063#	8661													
.\$STRAP	1063#	9367													
.\$STYPD	1063#	8963													
.\$STYPE	1063#	8816													
.\$STYPO	1063#	8886													

B01

MC-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1 MACY11 27(1006) 02-NOV-76 18:42 PAGE 210
DZRJEA.CMB 02-NOV-76 18:40 CROSS REFERENCE TABLE -- MACRO NAMES

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

NOV SEQ/SOL/CRF/LI:ME/NL:TOC:MC:MD:CND=DZRJEA.CMB
RUN-TIME: 121 139 15 SECONDS
RUN-TIME RATIO: 1656 276=5.9
CORE USED: 32K (.63 PAGES)

