

RM03,02

DUAL PORT LOGIC 1
CZRMGB0

AH-B010B-MC

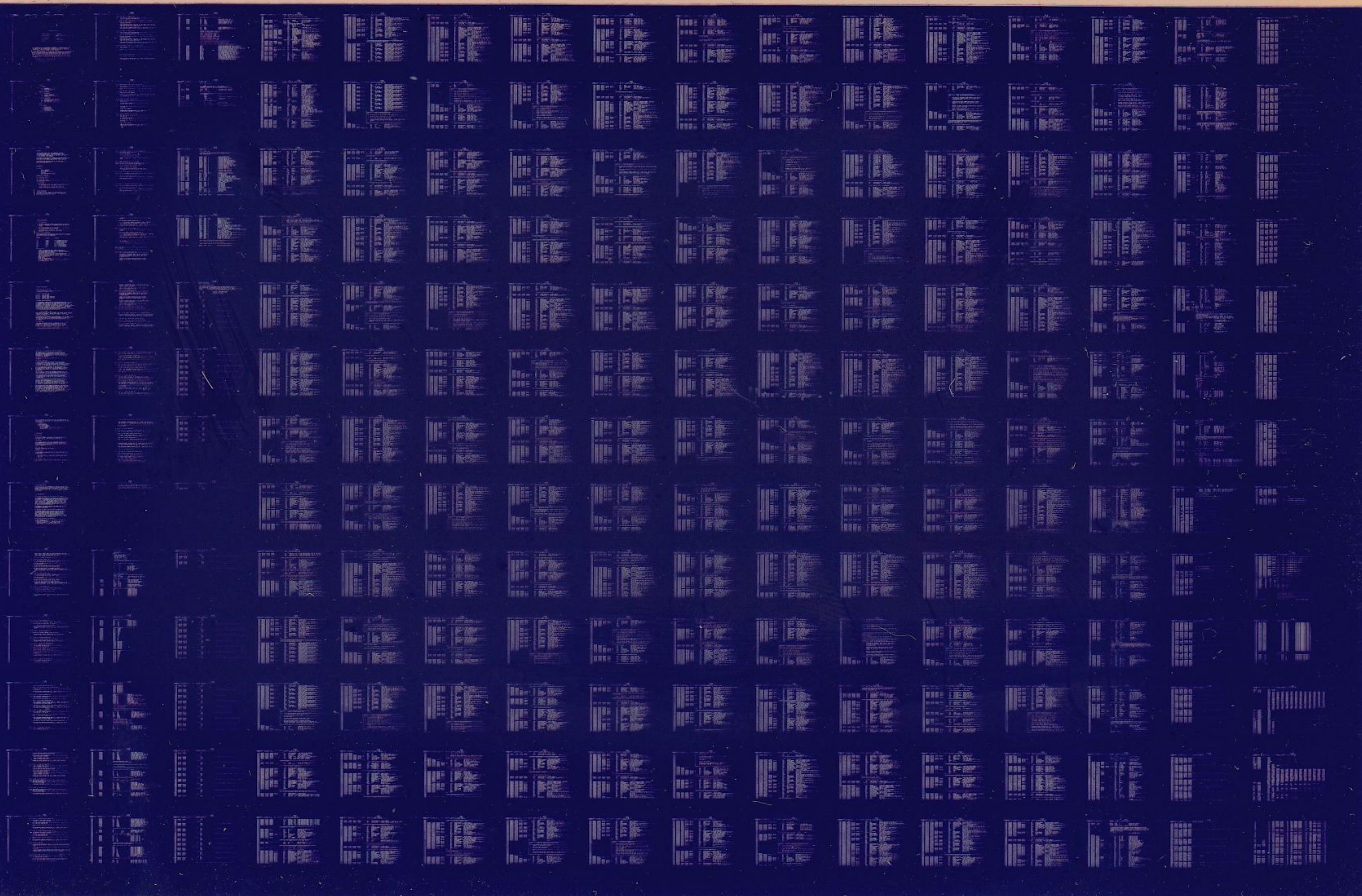
JAN 1978

COPYRIGHT © 1977

digital

FICHE 1 OF 2

MADE IN USA



RM03,02

DUAL PORT LOGIC 1
CZRMGB0

AH-B010B-MC

COPYRIGHT © 1977

FICHE 2 OF 2

JAN 1978

digital

MADE IN USA

[Faint, illegible text visible on the left edge of the page, likely bleed-through from the reverse side.]

.REM ↑

IDENTIFICATION

PRODUCT CODE: AC-B0098-MC
 PRODUCT NAME: CZRMGB0 RM03/RM02 DUAL PORT LOGIC TEST,
 PART 1
 DATE CREATED: 15 AUG 77
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: DOUG RIIKONEN

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

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

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

COPYRIGHT (C) 1977, DIGITAL EQUIPMENT CORPORATION

RM03 DUAL PORT LOGIC TEST, PART 1

CONTENTS

CZRMGB0 RM03/RM02 DUAL PORT LOGIC TEST, PART 1

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

1. ABSTRACT

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

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 PREREQUISITE PROGRAMS

RMO3 DISKLESS DIAGNOSTIC

RMO3 FUNCTIONAL TEST

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

2.3 OTHER PROGRAMS

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

B. DYNAMIC OPERATION OF THE DUAL PORT OPTION IS TESTED BY THE RMO3 PERFORMANCE EXERCISE PROGRAM.

3. LOADING PROCEDURES

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

153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
2084. STARTING PROCEDURES

4.1 STARTING ADDRESSES

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

4.2 UNIBUS & VECTOR ADDRESSES

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

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

4.3 OPERATOR ACTION

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

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

5. OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

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

THE SWITCH SETTINGS ARE:

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

5.2 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RM03 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

5.3 TEST SELECTION

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

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

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

PROGRAM WILL THEN EXECUTE ALL TESTS IN SEQUENCE.

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

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

5.4 TEST CABLE CONNECTION

TO TEST THE RMO3 DUAL PORT OPTION WITH THIS PROGRAM, A SPECIAL TEST CABLE MUST BE USED. (THE TEST CABLE IS P/N 7010507-02). THE TEST CABLE CONNECTS MASSBUS A & MASSBUS B TOGETHER AT THE DRIVE BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

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

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

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

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

WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION BITS FOR PORTS A & B ARE ASSERTED IN THE SAME BIT POSITION WHEN 'RMAS' (ATTENTION SUMMARY REGISTER) IS READ. THE ATTENTION BIT POSITION IS DETERMINED BY THE ADDRESS OF THE DRIVE THE ATTENTION BIT THAT APPEARS FOR THE DRIVE IS THE INCLUSIVE 'OR' OF THE PORT A & PORT B ATTENTION BITS. BECAUSE OF THIS, THE PROGRAM LOOKS AT ONLY THE ATTENTION BIT IN 'RMD5' (DRIVE STATUS REGISTER) TO DETERMINE THE STATE OF THE SELECTED PORTS'S ATTENTION BIT.

6. ERRORS

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

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

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

7. MISCELLANEOUS

7.1 RESTRICTIONS

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

7.2 LIMITATIONS

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

7.3 EXECUTION TIME

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

7.4 REQUIRED TESTS

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

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

7.5 DISK SURFACE USAGE

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

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

7.6 LOOP ON ERROR OPTION

IF SW<09> IS SET, THE PROGRAM WILL LOOP ON A FAILING TEST UNTIL EITHER THE SWITCH IS RESET OR THE ERROR STOPS OCCURRING. BECAUSE THE PROGRAM MUST RESET THE RMO3 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR WAS DETECTED.

8. TEST DESCRIPTIONS

8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

THE PROGRAM DETERMINES THAT THE DRIVE IS IN NEUTRAL BY CHECKING THE CONTENTS OF THE DRIVE STATUS REGISTER (RMO51) 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 RMO51, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DISCREPANCY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

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

8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

THE PROGRAM VERIFIES THAT THE DRIVE HAS BEEN SEIZED BY CHECKING THE DRIVE STATUS REGISTER (RMO51) THROUGH THE SEIZING PORT AND VERIFYING THAT CORRECT STATUS IS SEEN. WHEN RMO51 IS READ THROUGH THE OPPOSITE PORT, ZEROS SHOULD BE SEEN. IF BOTH CONDITIONS EXIST, (I.E. CORRECT STATUS THROUGH THE SEIZING PORT AND ZEROS THROUGH THE OPPOSITE PORT), THE PROGRAM CONCLUDES THAT THE DRIVE HAS BEEN SEIZED BY THE SPECIFIED PORT.

8.3 METHOD USED TO VERIFY PORT REQUESTS

THE PORT REQUEST FLOPS IN THE MAINTENANCE REGISTER ARE TESTED TO DETERMINE IF :
· A DRIVE IS IN NEUTRAL, I.E., RQA AND RQB ARE ZERO;
· A DRIVE IS SEIZED, I.E., RQA OR RQB IS ONE;
· A PORT REQUEST IS SET WHILE THE DRIVE IS SEIZED TO THE ALTERNATE PORT, I.E., RQA AND RQB ARE ONE.

TEST 1 NEUTRAL ACCESS TEST

VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS

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

- A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RMO3, THAT THE DRIVE IS ONLINE (RMDS1 HAS 'MOL' 'PGM' 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.
- B. THE TEST IS REPEATED THROUGH BOTH PORTS.

TEST 2 PORT 'A' SEIZE/TIMEOUT TEST

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

- A. WRITE 0'S INTO RML9 THROUGH PORT 'A'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B'; VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
- C. WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE. MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.

TEST 3 PORT 'B' SEIZE/TIMEOUT TEST

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

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

TEST 4 PORT 'A' SEIZE/RELEASE TEST

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
- B. SET VOLUME VALID AND CLEAR ANY ERROR
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.

489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

TEST 5 PORT 'B' SEIZE/RELEASE TEST

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- B. SET VOLUME VALID AND CLEAR ANY ERROR
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.

TEST 6 PORT 'A' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 7 PORT 'B' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 10 PORT 'A' RELEASE INTERFERENCE TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
- C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'.
- E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 11 PORT 'B' RELEASE INTERFERENCE TEST

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

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

545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

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

TEST 12 PORT 'A' RELEASE W/ERRORS TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
- B. WRITE 1'S INTO RMR1 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 RMR1 HAS NOT BEEN CLEARED.
- D. CLEAR RMR1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 13 PORT 'B' RELEASE W/ERRORS TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
- B. WRITE 1'S INTO RMR1 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 RMR1 HAS NOT BEEN CLEARED.
- D. CLEAR RMR1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 14 PORT 'A' SEIZE AND CLEAR TEST

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

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

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

TEST 15 PORT 'B' SEIZE AND CLEAR TEST

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

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

TEST 16 SEIZE 'A' BY RMCS1 TEST

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

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

TEST 17 SEIZE 'B' BY RMCS1 TEST

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

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

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

TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST

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

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

TEST 21 PORT 'B' INHIBIT SEIZE BY RMCS1 TEST

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

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

TEST 22 SEIZE BY RMAS TEST

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

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

TEST 23 INHIBIT SEIZE BY RMAS TEST

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

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

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

TEST 24 SET PORT 'A' REQUEST TEST

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

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

TEST 25 SET PORT 'B' REQUEST TEST

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

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

TEST 26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR

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

- A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
- C. ISSUE A DRIVE CLEAR COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

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

TEST 27 TEST RESET ATTENTION 'B' BY DRIVE CLEAR

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

- A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
- C. ISSUE A DRIVE CLEAR COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

TEST 30 RESET ATTENTION 'A' BY GO TEST

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

- A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH ATTENTION BITS ARE SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD5.
- C. ISSUE A NOP COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS RESET, AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

TEST 31 RESET ATTENTION 'B' BY GO TEST

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

- A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH ATTENTION BITS ARE SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD5.
- C. ISSUE A NOP COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS RESET, AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

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

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

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

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

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

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

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

TEST 34 PORT 'A' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'A'.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'B'.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 36 SET ATTENTION 'A' BY COMMAND TEST

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

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

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

TEST 37 SET ATTENTION 'B' BY COMMAND TEST

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

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

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

THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.

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

THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"

TEST 40 PORT 'A' SET VOLUME VALID TEST

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

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

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

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

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

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

A. WITH PORT 'B' SELECTED, RESET AND SET "UNIT READY" STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE IS SEIZED AND THAT "VOLUME VALID" IS RESET AND ATTENTION IS SET.

B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A. VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID IS SET.

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

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

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

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

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

B. WRITE 1'S INTO RMR1 THROUGH PORT 'A' TO FORCE AN ATTENTION.

C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND NOT SET FOR PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

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

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

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

B. WRITE 1'S INTO RMR1 THROUGH PORT 'B'.

C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

TEST 44 PORT 'A' RETRIGGER BY DEMAND TEST

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

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

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
- B. WAIT 500 MS AND READ RMDS1 THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
- D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 45 PORT 'B' RETRIGGER BY DEMAND TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- B. WAIT 500 MS AND WRITE 0'B INTO RMDS1 THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
- D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 46 PORT 'A' TIMEOUT/RELEASE TEST

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMDS1 FROM PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.
- E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 47 PORT 'B' TIMEOUT/RELEASE TEST

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

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

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

- B. SET PORT REQUEST BY WRITING 0'S INTO RMDS1 FROM PORT 'B'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

TEST 50 PORT 'A' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
- B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.
- C. READ RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RMER1, RMER2 THROUGH PORT 'A'.
- E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
- F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
- G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 51 PORT 'B' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
- B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.
- C. READ RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RMER1, RMER2 THROUGH PORT 'B'.
- E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
- F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS

1105
1106
1107
1108
1109
1110
1111
1112

SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.

G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

:PROGRAM REVISION #001

.TITLE CZRMGB0 RM03/2 DU POR LGC 1

.*COPYRIGHT (C) 1977

.*DIGITAL EQUIPMENT CORP.

.*MAYNARD, MASS. 01754

.*

.*PROGRAM BY D. RIIKONEN

.*

.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC

.*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

.*

.SBTTL OPERATIONAL SWITCH SETTINGS

.*

.* SWITCH

USE

.*

.*

.*

.*

.*

.*

.*

.*

15 HALT ON ERROR
14 LOOP ON TEST
13 INHIBIT ERROR TYPEOUTS
11 INHIBIT ITERATIONS
10 BELL ON ERROR
9 LOOP ON ERROR

.SBTTL BASIC DEFINITIONS

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

STACK= 1100

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

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

.*MISCELLANEOUS DEFINITIONS

HT= 11 ;:CODE FOR HORIZONTAL TAB

LF= 12 ;:CODE FOR LINE FEED

CR= 15 ;:CODE FOR CARRIAGE RETURN

CRLF= 200 ;:CODE FOR CARRIAGE RETURN-LINE FEED

PS= 177776 ;:PROCESSOR STATUS WORD

.EQUIV PS,PSW

STKLMT= 177774 ;:STACK LIMIT REGISTER

PIRQ= 177772 ;:PROGRAM INTERRUPT REQUEST REGISTER

DSWR= 177570 ;:HARDWARE SWITCH REGISTER

DDISP= 177570 ;:HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS

R0= %0 ;:GENERAL REGISTER

R1= %1 ;:GENERAL REGISTER

R2= %2 ;:GENERAL REGISTER

R3= %3 ;:GENERAL REGISTER

R4= %4 ;:GENERAL REGISTER

R5= %5 ;:GENERAL REGISTER

R6= %6 ;:GENERAL REGISTER

R7= %7 ;:GENERAL REGISTER

SP= %6 ;:STACK POINTER

PC= %7 ;:PROGRAM COUNTER

001100

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

```

1169
1170
1171      000000
1172      000040
1173      000100
1174      000140
1175      000200
1176      000240
1177      000300
1178      000340
1179
1180
1181      100000
1182      040000
1183      020000
1184      010000
1185      004000
1186      002000
1187      001000
1188      000400
1189      000200
1190      000100
1191      000040
1192      000020
1193      000010
1194      000004
1195      000002
1196      000001
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209      100000
1210      040000
1211      020000
1212      010000
1213      004000
1214      002000
1215      001000
1216      000400
1217      000200
1218      000100
1219      000040
1220      000020
1221      000010
1222      000004
1223      000002
1224      000001

.*PRIORITY LEVEL DEFINITIONS
PR0=      0          ;: PRIORITY LEVEL 0
PR1=     40          ;: PRIORITY LEVEL 1
PR2=    100          ;: PRIORITY LEVEL 2
PR3=    140          ;: PRIORITY LEVEL 3
PR4=    200          ;: PRIORITY LEVEL 4
PR5=    240          ;: PRIORITY LEVEL 5
PR6=    300          ;: PRIORITY LEVEL 6
PR7=    340          ;: PRIORITY LEVEL 7

.*"SWITCH REGISTER" SWITCH DEFINITIONS
SW15=    100000
SW14=    40000
SW13=    20000
SW12=    10000
SW11=    4000
SW10=    2000
SW09=    1000
SW08=    400
SW07=    200
SW06=    100
SW05=    40
SW04=    20
SW03=    10
SW02=    4
SW01=    2
SW00=    1
.EQUIV   SW09, SW9
.EQUIV   SW08, SW8
.EQUIV   SW07, SW7
.EQUIV   SW06, SW6
.EQUIV   SW05, SW5
.EQUIV   SW04, SW4
.EQUIV   SW03, SW3
.EQUIV   SW02, SW2
.EQUIV   SW01, SW1
.EQUIV   SW00, SW0

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

```


BASIC DEFINITIONS

```

1225 .EQUIV BIT09,BIT9
1226 .EQUIV BIT08,BIT8
1227 .EQUIV BIT07,BIT7
1228 .EQUIV BIT06,BIT6
1229 .EQUIV BIT05,BIT5
1230 .EQUIV BIT04,BIT4
1231 .EQUIV BIT03,BIT3
1232 .EQUIV BIT02,BIT2
1233 .EQUIV BIT01,BIT1
1234 .EQUIV BIT00,BIT0
1235
1236 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1237 000004 ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
1238 000010 RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
1239 000014 TBITVEC=14 ;: "T" BIT
1240 000014 TRIVEC= 14 ;: TRACE TRAP
1241 000014 BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
1242 000020 IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
1243 000024 PWRVEC= 24 ;: POWER FAIL
1244 000030 EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
1245 000034 TRAPVEC=34 ;: "TRAP" TRAP
1246 000060 TKVEC= 60 ;: TTY KEYBOARD VECTOR
1247 000064 TPVEC= 64 ;: TTY PRINTER VECTOR
1248 000240 PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR
1249
1250 ;:*****
1251
1252 .SBTTL RH70 REGISTERS
1253
1254 ;:*****
1255
1256 ;CONTROL AND STATUS REGISTER 1 (RMCS1)
1257
1258 000100 IE= 100 ;: INTERRUPT ENABLE (BIT #6)
1259 000200 RDY= 200 ;: READY (BIT #7)
1260 000400 A16= 400 ;: HIGH ORDER BUS ADDRESS BIT (BIT #8)
1261 001000 A17= 1000 ;: HIGH ORDER BUS ADDRESS BIT (BIT #9)
1262 002000 PSEL= 2000 ;: PORT SELECT (BIT #10)
1263 020000 MCPE= 20000 ;: MASSBUSS PARITY ERROR (BIT #13)
1264 040000 TRE= 40000 ;: TRANSFER ERROR (BIT #14)
1265 100000 SC= 100000 ;: SPECIAL CONDITION (BIT #15)
1266
1267 ;WORD COUNT REGISTER (RMWC)
1268 ;(EACH BIT IS CALLED BY BIT NUMBER)
1269
1270 ;BUS ADDRESS REGISTER (RMBA)
1271 ;(EACH BIT IS CALLED BY BIT NUMBER)
1272
1273 ;CONTROL AND STATUS REGISTER 2 (RMCS2)
1274
1275 000001 U0= 1 ;: UNIT SELECT (BIT #0)
1276 000002 U1= 2 ;: UNIT SELECT (BIT #1)
1277 000004 U3= 4 ;: UNIT SELECT (BIT #2)
1278 000010 BAI= 10 ;: BUS ADDRESS INCREMENT INHIBIT (BIT #3)
1279 000020 PAT= 20 ;: MASSBUS PARITY TEST (BIT #4)
1280 000040 CLR= 40 ;: CLEAR (BIT #5)

```

1281 000100
1282 000200
1283 000400
1284 001000
1285 002000
1286 004000
1287 010000
1288 020000
1289 040000
1290 100000

IR= 100 ; INPUT READY (BIT #6)
OR= 200 ; OUTPUT READY (BIT #7)
MOPE= 400 ; MASS BUS PARITY ERROR (BIT #8)
MXF= 1000 ; MISSED TRANSFER ERROR (BIT #9)
PGE= 2000 ; PROGRAM ERROR (BIT #10)
NEM= 4000 ; NON EXISTENT MEMORY (BIT #11)
NED= 10000 ; NON EXISTENT DRIVE (BIT #12)
UPE= 20000 ; UNIBUS PARITY ERROR (BIT #13)
WCE= 40000 ; WRITE CHECK ERROR (BIT #14)
DLT= 100000 ; DATA LATE (BIT #15)

; DATA BUFFER REGISTER (RMD8)
;(EACH BIT IS CALLED BY BIT NUMBER)

;*****

.SBTTL RMO3 REGISTERS

;*****

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

1304 000001
1305 000002
1306 000004
1307 000010
1308 000020
1309 000040
1310 004000

GO= 1 ; GO BIT (BIT #0)
FO= 2 ; FUNCTION CODE BIT #1
F1= 4 ; FUNCTION CODE BIT #2
F2= 10 ; FUNCTION CODE BIT #3
F3= 20 ; FUNCTION CODE BIT #4
F4= 40 ; FUNCTION CODE BIT #5
DVA= 4000 ; DEVICE AVAILABLE (BIT #11)

; CONTROL STATUS REGISTER #2 (RMCS2)

1314 000040

CLR= BITS ; CONTROLLER CLEAR

; DRIVE STATUS REGISTER (RMD51) (#01)

1318 000001
1319 000100
1320 000200
1321 000400
1322 001000
1323 002000
1324 004000
1325 010000
1326 020000
1327 040000
1328 100000

OM = BIT00 ; OFFSET MODE
VV= 100 ; VOLUME VALID (BIT #6)
DRY= 200 ; DRIVE READY (BIT #7)
DPR= 400 ; DRIVE PRESENT (BIT #8)
PGM= 1000 ; PROGRAMABLE (BIT #9)
LBT= 2000 ; LAST SECTOR TRANSFERRED (BIT #10)
WRL= 4000 ; WRITE LOCK (BIT #11)
MOL= 10000 ; MEDIUM ON-LINE (BIT #12)
PIP= 20000 ; POSITIONING OPERATION IN PROGRESS (BIT #13)
ERR= 40000 ; COMPOSITE ERROR (BIT #14)
ATA= 100000 ; ATTENTION ACTIVE (BIT #15)

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

1332 000001
1333 000002
1334 000004
1335 000010
1336 000020

ILF= 1 ; ILLEGAL FUNCTION (BIT #0)
ILR= 2 ; ILLEGAL REGISTER (BIT #1)
RMR= 4 ; REGISTER MODIFICATION REFUSED (BIT #2)
PAR= 10 ; PARITY ERROR (BIT #3)
FER= 20 ; FORMAT ERROR (BIT #4)

1337	000040	WCF=	40	;WRITE CLOCK FAIL (BIT #5)
1338	000100	ECH=	100	;ECC HARD ERROR (BIT #6)
1339	000200	HCE=	200	;HEADER COMPARE ERROR (BIT #7)
1340	000400	HCRC=	400	;HEADER CRC ERROR (BIT #8)
1341	001000	AOE=	1000	;ADDRESS OVERFLOW ERROR (BIT #9)
1342	002000	IAE=	2000	;INVALID ADDRESS ERROR (BIT #10)
1343	004000	WLE=	4000	;WRITE LOCK ERROR (BIT #11)
1344	010000	DTE=	10000	;DRIVE TIMING ERROR (BIT #12)
1345	020000	OPI=	20000	;OPERATION INCOMPLETE (BIT #13)
1346	040000	UNS=	40000	;DRIVE UNSAFE (BIT #14)
1347	100000	DCK=	100000	;DATA CHECK ERROR (BIT 15)
1348				
1349				;MAINTAINABILITY REGISTER (RMMR1)(#03)
1350				
1351	000001	DMD=	1	;DIAGINOSTIC MODE (BIT #0)
1352	001000	MUR	= BIT09	;MAINTENANCE UNIT READY
1353	040000	RQB	= BIT14	;PORT B REQUEST FLOP
1354	100000	RQA	= BIT15	;PORT A REQUEST FLOP
1355				
1356				;ATTENTION SUMMARY PSEUDO-REGISTER (RMAS) (#04)
1357				
1358	000001	AT0=	1	;DEVICE 0 (BIT #0)
1359	000002	AT1=	2	;DEVICE 1 (BIT #1)
1360	000004	AT2=	4	;DEVICE 2 (BIT #2)
1361	000010	AT3=	10	;DEVICE 3 (BIT #3)
1362	000020	AT4=	20	;DEVICE 4 (BIT #4)
1363	000040	AT5=	40	;DEVICE 5 (BIT #5)
1364	000100	AT6=	100	;DEVICE 6 (BIT #6)
1365	000200	AT7=	200	;DEVICE 7 (BIT #7)
1366				
1367				;DESIRED SECTOR/TRACK ADDRESS REGISTER (RMDA) (#05)
1368				; (EACH BIT IS CALLED BY BIT NUMBER)
1369				
1370				;DRIVE TYPE REGISTER (RMDT) (#06)
1371				
1372	000001	DT00=	1	;DRIVE TYPE NUMBER BIT 1
1373	000002	DT01=	2	;DRIVE TYPE NUMBER BIT 2
1374	000004	DT02=	4	;DRIVE TYPE NUMBER BIT 3
1375	000010	DT03=	10	;DRIVE TYPE NUMBER BIT 4
1376	000020	DT04=	20	;DRIVE TYPE NUMBER BIT 5
1377	000040	DT05=	40	;DRIVE TYPE NUMBER BIT 6
1378	000100	DT06=	100	;DRIVE TYPE NUMBER BIT 7
1379	000200	DT07=	200	;DRIVE TYPE NUMBER BIT 8
1380	000400	DT08=	400	;DRIVE TYPE NUMBER BIT 9
1381	004000	DRQ=	4000	;DRIVE REQUEST REQUIRED (BIT #11)
1382	020000	MOH=	20000	;MOVING HEAD (BIT #13)
1383	040000	TAP=	40000	;TAPE DRIVE (BIT #14)
1384	100000	NBA=	100000	;NOT BLOCK ADDRESSED (BIT #15)
1385				
1386				;LOOK-AHEAD REGISTER (RMLA) (#07)
1387				
1388	000100	SC0=	100	;SECTOR COUNT FIELD 0 (BIT #6)
1389	000200	SC1=	200	;SECTOR COUNT FIELD 1 (BIT #7)
1390	000400	SC2=	400	;SECTOR COUNT FIELD 2 (BIT #8)
1391	001000	SC3=	1000	;SECTOR COUNT FIELD 3 (BIT #9)
1392	002000	SC4=	2000	;SECTOR COUNT FIELD 4 (BIT #10)

1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448

000010
000700
002L00
004000
010000
020000
100000

000200
002000
004000
010000

000000
000002
000004
000006
000010
000012
000014
000016
000020
000022
000024
000026
000030
000032
000034
000040
000042
000044
000046

;RM03 ERROR REGISTER #2 (RMER2) (#10)

DPE= 10 ;DATA PARITY ERROR (BIT #3)
DVC= 200 ;DEVICE CHECK (BIT #7)
LBC= 2000 ;LOSS OF BIT CLOCK (BIT #10)
LSC= 4000 ;LOSS OF SYSTEM CLOCK (BIT #11)
IVC= 10000 ;INVALID COMMAND (BIT #12)
OPE= 20000 ;OPERATOR ERROR (BIT #13)
SKI= 100000 ;SEEK INCOMPLETE (BIT #14)

;OFFSET REGISTER (RMOF) (#11)

OFD= 200 ;OFFSET FORWARD (BIT #5)
HCI= 2000 ;HEADER COMPARE INHIBIT (BIT #10)
ECI= 4000 ;ERROR CORRECTION CODE INHIBIT (BIT #11)
FMT16= 10000 ;FORMAT BIT (BIT #12)

;DESIRED CYLINDER ADDRESS (RMDC) (#12)
;(EACH BIT IS CALLED BY BIT NUMBER)

;SERIAL NUMBER REGISTER (RMSN) (#14)
;(EACH IS CALLED BY BIT NUMBER)

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

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

;*****

.SB/TL DEFINITIONS OF THE RH70/RM03 ADDRESS INDEXES

;*****

RMCS1=0 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
RMLC=2 ;WORD COUNT REGISTER (NOT A DRIVE REG)
RMBA=4 ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
RMDA=6 ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
RMCS2=10 ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
RMDS1=12 ;DRIVE STATUS REGISTER (DRIVE REG 01)
RMER1=14 ;ERROR REGISTER #1 (DRIVE REG. 02)
RMA5=16 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
RMLA=20 ;LOCK AHEAD REGISTER (DRIVE REG. 07)
RMD8=22 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
RMMA1=24 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
RMDT=26 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
RMSN=30 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
RMOF=32 ;OFFSET REGISTER (DRIVE REG. 11)
RMDC=34 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
RMMA2=40 ;MAINTENANCE REGISTER #2 (DRIVE REG. 14)
RMER2=42 ;ERROR REGISTER #2 (DRIVE REG. 15)
RMEC1=44 ;ECC POSITION REGISTER (DRIVE REG. 16)
RMEC2=46 ;ECC PATTERN REGISTER (DRIVE REG. 17)


```

1449      .SBTTL  TRAP CATCHER
1450
1451      000000      .=0
1452      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1453      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1454      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1455      000174      .=174
1456      000174      000000      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
1457      000176      000000      SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
1458
1459      .SBTTL  ACT11 HOOKS
1460
1461      ;*****
1462      ;HOOKS REQUIRED BY ACT11
1463      $SVPC=.      ;SAVE PC
1464      .=46
1465      000046      065746      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
1466      000052      000052      .=52
1467      000052      020000      .WORD 20000      ;;2)SET LOC.52 TO 20000
1468      000200      000200      .=$SVPC      ;;RESTORE PC
1469
1470      .SBTTL  STARTING ADDRESS = 200
1471
1472      000200      000137      002160      JMP      START      ;START THE PROGRAM
1473
1474      .SBTTL  START THE PROGRAM AND CHANGE THE RH70 ADDRESS = 204
1475
1476      000204      000137      002166      JMP      START1     ;START AND CHANGE THE RH70 ADDRESS
1477
1478

```

```

1479
1480
1481
1482
1483
1484
1485      001100
1486      001100      000000
1487      001100      000
1488      001102      000
1489      001103      000
1490      001104      000000
1491      001106      000000
1492      001110      000000
1493      001112      000000
1494      001114      000
1495      001115      001
1496      001116      000000
1497      001120      000000
1498      001122      000000
1499      001124      000000
1500      001126      000000
1501      001130      000000
1502      001132      000000
1503      001134      000
1504      001135      000
1505      001136      000000
1506      001140      177570
1507      001142      177570
1508      001144      177560
1509      001146      177562
1510      001150      177564
1511      001152      177566
1512      001154      000
1513      001155      002
1514      001156      012
1515      001157      000
1516      001160      000000
1517
1518      001162      000000
1519      001164      000000
1520      001166      000000
1521      001170      000000
1522      001172      000000
1523      001174      000000
1524      001176      000000
1525      001200      000000
1526      001202      177607      000377
1527      001206      077
1528      001207      015
1529      001210      000012
1530
1531      000015
1532      000012
1533      001212      172540
1534      001214      172542

```

.SBTTL COMMON TAGS

```

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

```

```

.=1100
$CMTAG:      .WORD      0      ; START OF COMMON TAGS
$PASS:      .WORD      0      ; CONTAINS PASS COUNT
$TSTNM:     .BYTE      0      ; CONTAINS THE TEST NUMBER
$ERFLG:     .BYTE      0      ; CONTAINS ERROR FLAG
$ICNT:      .WORD      0      ; CONTAINS SUBTEST ITERATION COUNT
$LPADR:     .WORD      0      ; CONTAINS SCOPE LOOP ADDRESS
$LPERR:     .WORD      0      ; CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL:     .WORD      0      ; CONTAINS TOTAL ERRORS DETECTED
$ITEMB:     .BYTE      0      ; CONTAINS ITEM CONTROL BYTE
$ERMAX:     .BYTE      1      ; CONTAINS MAX. ERRORS PER TEST
$ER3PC:     .WORD      0      ; CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR:     .WORD      0      ; CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR:     .WORD      0      ; CONTAINS ADDRESS OF 'BAD' DATA
$GGDAT:     .WORD      0      ; CONTAINS 'GOOD' DATA
$BDCAT:     .WORD      0      ; CONTAINS 'BAD' DATA
              .WORD      0      ; RESERVED--NOT TO BE USED
$AUTOB:     .BYTE      0      ; AUTOMATIC MODE INDICATOR
$INTAG:     .BYTE      0      ; INTERRUPT MODE INDICATOR
              .WORD      0
$SWR:       .WORD      0      ; ADDRESS OF SWITCH REGISTER
$DISP:      .WORD      0      ; ADDRESS OF DISPLAY REGISTER
$TKS:       .WORD      0      ; TTY KBD STATUS
$TKB:       .WORD      0      ; TTY KBD BUFFER
$TPS:       .WORD      0      ; TTY PRINTER STATUS REG. ADDRESS
$TPB:       .WORD      0      ; TTY PRINTER BUFFER REG. ADDRESS
$NULL:      .BYTE      0      ; CONTAINS NULL CHARACTER FOR FILLS
$FILLS:     .BYTE      2      ; CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC:     .BYTE      12     ; INSERT FILL CHARS. AFTER A "LINE FEED"
$STPFLG:    .BYTE      0      ; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$REGAD:     .WORD      0      ; CONTAINS THE ADDRESS FROM WHICH ($REGO) WAS OBTAINED
$REGO:      .WORD      0      ; CONTAINS (($REGAD)+0)
$TMP0:      .WORD      0      ; USER DEFINED
$TMP1:      .WORD      0      ; USER DEFINED
$TMP2:      .WORD      0      ; USER DEFINED
$TMP3:      .WORD      0      ; USER DEFINED
$TMP4:      .WORD      0      ; USER DEFINED
$TIMES:     .WORD      0      ; MAX. NUMBER OF ITERATIONS
$ESCAPE:    .WORD      0      ; ESCAPE ON ERROR ADDRESS
$BELL:      .ASCIZ    <207><377><377> ; CODE FOR BELL
$QUES:      .ASCIZ    /?/         ; QUESTION MARK
$CRLF:      .ASCIZ    <15>        ; CARRIAGE RETURN
$LF:        .ASCIZ    <12>        ; LINE FEED
*****
CR          =          15
LF          =          12
$LKCSR:     .WORD      172540     ; ADDR OF KW11-P STATUS REGISTER
$LKCSB:     .WORD      172542     ; ADDR OF KW11-P COUNTER BUFFER

```

1535	001216	000104	\$LPVEC: .WORD	104	: ADDR OF KW11-P VECTOR
1536	001220	177546	\$LKS: .WORD	177546	: ADDR OF KW11-L STATUS REGISTER
1537	001222	000100	\$LLVEC: .WORD	100	: ADDR OF KW11-L VECTOR
1538	001224	000000	PORTA: .WORD	0	: ADDRESS OF PORT A
1539	001226	000000	PORTB: .WORD	0	: ADDRESS OF PORT B
1540	001230	000000	PORTC: .WORD	0	: ADDRESS OF DIFFERENT DRIVE
1541	001232	000000	RQSTA: .WORD	0	: REQUEST BIT FOR PORT A
1542	001234	000000	RQSTB: .WORD	0	: REQUEST BIT FOR PORT B
1543	001236	000000	ASR1: .WORD	0	: ATA-A OR ATA-B = 1
1544	001240	000000	PTNBR: .WORD	0	: CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
1545	001242	000000	SEIZPT: .WORD	0	: CONTAINS THE ADDRESS OF THE SEIZING PORT
1546	001244	000000	OPPRT: .WORD	0	: CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
1547	001246	000000	TSTNUM: .WORD	0	: NUMBER OF THE CURRENT TEST
1548	001250	000000	CKERR: .WORD	0	: IF -1, A REGISTER MISCOMPARISON OCCURRED
1549	001252	000000	NOSEIZ: .WORD	0	: IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
1550	001254	000000	RELERR: .WORD	0	: IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
1551	001256	000000	TIME: .WORD	0	: ELAPSED TIME COUNTER
1552	001260	000000	WATCH: .WORD	0	: WATCH DOG TIMER LOCATION
1553	001262	000000	TIMEA: .WORD	0	: THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
1554	001264	000000	TIMEAP: .WORD	0	: PORT A TIMEOUT VALUE + 25%
1555	001266	000000	TIMEAM: .WORD	0	: PORT A TIMEOUT VALUE - 25%
1556	001270	000000	TIMEB: .WORD	0	: THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
1557	001272	000000	TIMEBP: .WORD	0	: PORT B TIMEOUT VALUE + 25%
1558	001274	000000	TIMEBM: .WORD	0	: PORT B TIME VALUE - 25%
1559	001276	000000	TIMES: .WORD	0	: STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
1560	001300	000000	KYBCTL: .WORD	0	: SINGLE TEST INDICATOR
1561	001302	000000	CHGADR: .WORD	0	: CHANGE THE RH70 ADDRESS INDICATOR
1562					
1563					:;*****
1564					
1565					.SBTTL RH70/RM03 UNIBUS AND VECTOR ADDRESSES
1566					
1567					:;*****
1568					
1569	001304	176700	\$RMADR: .WORD	176700	: RH70/RM03 UNIBUS ADDRESS
1570	001306	000254	\$RMVEC: .WORD	254	: RH70 INTERRUPT VECTOR ADDRESS
1571					

1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
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

.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
    
```

001310

001310 072026
001312 076447
001314 100340
001316 100626

001320 072047
001322 076447
001324 100340
001326 100626

001330 072071
001332 076520
001334 100354
001336 100626

001340 072153
001342 076567
001344 100422
001346 100641

001350 072204
001352 076712
001354 100370
001356 100633

001360 072252
001362 077162
001364 100442

1628	001366	100633	DF5	
1629				
1630				;ERROR 7
1631				
1632	001370	072352	EM7	;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1633	001372	076766	DH7	
1634	001374	100422	DT7	
1635	001376	100641	DF7	
1636				
1637				;ERROR 10
1638				
1639	001400	072433	EM10	;REGISTER CONTENTS INCORRECT
1640	001402	076712	DH5	
1641	001404	100370	DT5	
1642	001406	100633	DF5	
1643				
1644				;ERROR 11
1645				
1646	001410	072463	EM11	;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1647	001412	077111	DH11	
1648	001414	100340	DT1	
1649	001416	100626	DF1	
1650				
1651				;ERROR 12
1652				
1653	001420	072547	EM12	;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1654	001422	077656	DH36	
1655	001424	100530	DT37	
1656	001426	100654	DF36	
1657				
1658				;ERROR 13
1659				
1660	001430	072617	EM13	; 'VOLUME VALID' BIT NOT SET BY READIN PRESET
1661	001432	077162	DH13	
1662	001434	100442	DT13	
1663	001436	100633	DF5	
1664				
1665				;ERROR 14
1666				
1667	001440	072704	EM14	; 'VOLUME VALID' SET ON THE OPPOSITE PORT
1668	001442	077162	DH13	
1669	001444	100442	DT13	
1670	001446	100633	DF5	
1671				
1672				;ERROR 15
1673				
1674	001450	072747	EM15	;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1675	001452	076766	DH7	
1676	001454	100422	DT7	
1677	001456	100641	DF7	
1678				
1679				;ERROR 16
1680				
1681	001460	073026	EM16	;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1682	001462	076766	DH7	
1683	001464	100422	DT7	

1684	001466	100641	DF7	
1685				
1686				;ERROR 17
1687				
1688	001470	073101	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
1689	001472	076766	DH7	
1690	001474	100422	DT7	
1691	001476	100641	DF7	
1692				
1693				;ERROR 20
1694				
1695	001500	073160	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
1696	001502	077656	DH36	
1697	001504	100530	DT37	
1698	001506	100654	DF36	
1699				
1700				;ERROR 21
1701				
1702	001510	073240	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
1703	001512	077656	DH36	
1704	001514	100530	DT37	
1705	001516	100654	DF36	
1706				
1707				;ERROR 22
1708				
1709	001520	073313	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT. REQUEST NOT SET
1710	001522	077302	DH22	
1711	001524	100460	DT22	
1712	001526	100650	DF31	

CZRMGB0 RM03-2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 34
ERROR POINTER TABLE

SEQ 0034

1713
1714
1715
1716
1717
1718

001530 073400
001532 077400
001534 100472

:ERROR 23

EM23
DH23
DT23

:TIMEOUT CLEARED THE DRIVE'S ERROR BIT

1719	001536	100626	DF1	
1720				
1721			:ERROR 24	
1722				
1723	001540	073446	EM24	:RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1724	001542	077400	DH23	
1725	001544	100472	DT23	
1726	001546	100626	DF1	
1727				
1728				
1729			:ERROR 25	
1730				
1731	001550	073525	EM25	:TIMEOUT ONE-SHOT DID NOT RETRIGGER
1732	001552	077656	DH36	
1733	001554	100520	DT36	
1734	001556	100654	DF36	
1735				
1736				
1737			:ERROR 26	
1738				
1739	001560	073570	EM26	:DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET

1740	001562	077302	DH22	
1741	001564	100460	DT22	
1742	001566	100650	DF31	
1743				
1744				;ERROR 27
1745				
1746	001570	073655	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1747	001572	076766	DH7	
1748	001574	100422	DT7	
1749	001576	100641	DF7	
1750				
1751				;ERROR 30
1752				
1753	001600	073733	EM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1754	001602	077656	DH36	
1755	001604	100520	DT36	
1756	001606	100654	DF36	
1757				
1758				;ERROR 31
1759				
1760	001610	074030	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SE
1761	001612	077557	DH31	
1762	001614	100506	DT31	
1763	001616	100650	DF31	
1764				
1765				;ERROR 32
1766				
1767	001620	074105	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1768	001622	076712	DH5	
1769	001624	100370	DT5	
1770	001626	100633	DF5	
1771				
1772				;ERROR 33
1773				
1774	001630	074156	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1775	001632	077656	DH36	
1776	001634	100520	DT36	
1777	001636	100654	DF36	
1778				
1779				;ERROR 34
1780				
1781	001640	074260	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1782	001642	077656	DH36	
1783	001644	100520	DT36	
1784	001646	100654	DF36	
1785				
1786				;ERROR 35
1787				
1788	001650	074363	EM35	;DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT 0
1789	001652	077656	DH36	

1790	001654	100530	DT37	
1791	001656	100654	DF36	
1792				
1793				; ERROR 36
1794				
1795	001660	074442	EM36	; TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
1796	001662	077656	DH36	
1797	001664	100520	DT36	
1798	001666	100654	DF36	
1799				
1800				; ERROR 37
1801				
1802	001670	074514	EM37	; DRIVE IS NON-EXISTENT
1803	001672	077656	DH36	
1804	001674	100530	DT37	
1805	001676	100654	DF36	
1806				
1807				; ERROR 40
1808				
1809	001700	074562	EM40	; ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1810	001702	076447	DH1	
1811	001704	100472	DT23	
1812	001706	100626	DF1	
1813				
1814				; ERROR 41
1815				
1816	001710	074637	EM41	; TIMEOUT CLEARED ATTENTION BIT
1817	001712	077400	DH23	
1818	001714	100472	DT23	
1819	001716	100626	DF1	
1820				
1821				; ERROR 42
1822				
1823	001720	074701	EM42	; DRIVE NOT IN NEUTRAL OR SEIZED
1824	001722	077705	DH42	
1825	001724	100540	DT42	
1826	001726	100657	DF42	
1827				
1828				; ERROR 43
1829				
1830	001730	074767	EM43	; DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1831	001732	077705	DH42	
1832	001734	100540	DT42	
1833	001736	100657	DF42	
1834				
1835				; ERROR 44
1836				
1837	001740	075044	EM44	; WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1838	001742	077724	DH44	
1839	001744	100506	DT31	
1840	001746	100650	DF31	
1841				
1842				; ERROR 45
1843				
1844	001750	075121	EM45	; CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A' 'B'
1845	001752	076447	DH1	

1846	001754	100340	DT1	
1847	001756	100626	DF1	
1848				
1849				
1850				; ERROR 46
1851	001760	075200	EM46	; CAN'T ACCESS DRIVE THROUGH EITHER PORT
1852	001762	100022	DH46	
1853	001764	100546	DT46	
1854	001766	100650	DF31	
1855				
1856				; ERROR 47
1857				
1858	001770	075247	EM47	; ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
1859	001772	077400	DH23	
1860	001774	100472	DT23	
1861	001776	100626	DF1	
1862				
1863				; ERROR 50
1864				
1865	002000	075335	EM50	; ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMA
1866	002002	077162	DH13	
1867	002004	100442	DT13	
1868	002006	100633	DF5	
1869				
1870				; ERROR 51
1871				
1872	002010	075417	EM51	; ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1873	002012	076712	DH5	
1874	002014	100370	DT5	
1875	002016	100633	DF5	
1876				
1877				; ERROR 52
1878				
1879	002020	075506	EM52	; ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1880	002022	077162	DH13	
1881	002024	100442	DT13	
1882	002026	100633	DF5	
1883				
1884				; ERROR 53
1885				
1886	002030	075601	EM53	; CAN'T READ ATTN BIT FROM OPPOSITE PORT
1887	002032	077400	DH23	
1888	002034	100340	DT1	
1889	002036	100626	DF1	
1890				
1891				; ERROR 54
1892				
1893	002040	075662	EM54	; RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1894	002042	077302	DH22	
1895	002044	100560	DT54	
1896	002046	100650	DF31	
1897				
1898				; ERROR 55
1899				
1900	002050	075755	EM55	; TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1901	002052	100120	DH55	

1902	002054	100572	DT55	
1903	002056	100661	DF55	
1904				
1905				;ERROR 56
1906				
1907	002060	076022	EM56	;RH11 DIDN'T RESPOND TO ADDRESSING
1908	002062	100176	DH56	
1909	002064	100604	DT56	
1910	002066	100665	DF56	
1911				
1912				
1913				;ERROR 57
1914	002070	076064	EM57	;PORT REQUEST FLOPS WRONG
1915	002072	100205	DH57	
1916	002074	100610	DT57	
1917	002076	100633	DF57	
1918				
1919				;ERROR 60
1920	002100	076125	EM60	;ATTENTION BITS NOT RESET BY RMAS
1921	002102	076712	DH5	
1922	002104	100370	DT5	
1923	002106	100633	DF5	
1924				
1925				;ERROR 61
1926	002110	076171	EM61	;ATTENTION NOT RESET BY GO
1927	002112	077400	DH23	
1928	002114	100472	DT23	
1929	002116	100626	DF1	
1930				
1931				;ERROR 62
1932	002120	076223	EM62	;ATTENTION RESET BY GO WHEN NOT SEIZED
1933	002122	077162	DH13	
1934	002124	100442	DT13	
1935	002126	100633	DF5	
1936				
1937				;ERROR 63
1938	002130	076271	EM63	;DRIVE SEIZED BY UNIT READY CHANGE
1939	002132	077656	DH36	
1940	002134	100520	DT36	
1941	002136	100654	DF36	
1942				
1943				;ERROR 64
1944	002140	076333	EM64	;ATTENTION NOT SET BY UNIT READY CHANGE
1945	002142	076766	DH7	
1946	002144	100422	DT7	
1947	002146	100641	DF7	
1948				
1949				;ERROR 65
1950	002150	075402	EM65	;VV NOT RESET BY UNIT READY
1951	002152	076712	DH5	
1952	002154	100370	DT5	
1953	002156	100633	DF5	
1954				
1955				
1956				;*****
1957				

```

1958 .SBTTL STARTUP AND INITIALIZATION ROUTINES
1959
1960 ;;*****
1961
1962 002160 005037 001302 START: CLR CHGADR ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
1963 002164 000403 BR START2 ;GO TO THE START
1964 002166 012737 177777 001302 START1: MOV #-1,CHGADR ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
1965 002174 000005 START2: RESET ;CLEAR THE BUS
1966 .SBTTL INITIALIZE THE COMMON TAGS
1967 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1968 002176 012706 001100 MOV $CMTAG,R6 ;FIRST LOCATION TO BE CLEARED
1969 002202 005026 CLR (R6)+ ;CLEAR MEMORY LOCATION
1970 002204 022706 001140 CMP $SWR,R6 ;;DONE?
1971 002210 001374 BNE -6 ;LOOP BACK IF NO
1972 002212 012706 001100 MOV $STACK,SP ;SETUP THE STACK POINTER
1973 ;;INITIALIZE A FEW VECTORS
1974 002216 012737 066222 000020 MOV $SCOPE,$IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
1975 002224 012737 060340 000022 MOV $340,$IOTVEC+2 ;LEVEL 7
1976 002232 012737 066454 000030 MOV $ERROR,$EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
1977 002240 012737 060340 000032 MOV $340,$EMTVEC+2 ;LEVEL 7
1978 002246 012737 071254 000034 MOV $STRAP,$TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
1979 002254 012737 060340 000036 MOV $340,$TRAPVEC+2 ;LEVEL 7
1980 002262 013737 065612 065604 MOV $ENDCT,$SEOPCT ;SETUP END-OF-PROGRAM COUNTER
1981 002270 005037 001176 CLR $TIMES ;INITIALIZE NUMBER OF ITERATIONS
1982 002274 005037 001200 CLR $ESCAPE ;CLEAR THE ESCAPE ON ERROR ADDRESS
1983 002300 012737 000001 001115 MOVB #1,$ERMAX ;ALLOW ONE ERROR PER TEST
1984 002306 012737 002306 001106 MOV $,$SLPADR ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1985 002314 012737 062314 001110 MOV $,$SLPERR ;SETUP THE ERROR LOOP ADDRESS
1986 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1987 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1988 002322 013746 000004 MOV $ERRVEC,-(SP) ;SAVE ERROR VECTOR
1989 002326 012737 002362 000004 MOV $64,$ERRVEC ;SET UP ERROR VECTOR
1990 002334 012737 177570 001140 MOV $DSWR,$SWR ;SETUP FOR A HARDWARE SWITCH REGISTER
1991 002342 012737 177570 001142 MOV $DDISP,$DISPLAY ;AND A HARDWARE DISPLAY REGISTER
1992 002350 022777 177777 176562 CMP #-1,$SWR ;TRY TO REFERENCE HARDWARE SWR
1993 002356 001012 BNE 66$ ;BRANCH IF NO TIMEOUT TRAP OCCURRED
1994 ;AND THE HARDWARE SWR IS NOT = -1
1995 002360 000403 BR 65$ ;BRANCH IF NO TIMEOUT
1996 002362 012716 002370 64$: MOV $65$,(SP) ;SET UP FOR TRAP RETURN
1997 002366 000002 RTI
1998 002370 012737 000176 001140 65$: MOV $SWREG,$SWR ;POINT TO SOFTWARE SWR
1999 002376 012737 000174 001142 MOV $DISPREG,$DISPLAY
2000 002404 012637 000004 66$: MOV (SP)+,$ERRVEC ;RESTORE ERROR VECTOR
2001
2002 002410 005227 177777 INC #-1 ;FIRST START ?
2003 002414 001002 BNE 1$ ;BR IF NOT
2004 002416 104401 071342 TYPE TITLE ;TYPE PROGRAM NAME
2005 002422 004737 067664 1$: JSR PC,$TKINT ;SETUP THE TTY KEYBOARD
2006 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
2007 002426 005737 000042 TST $42 ;ARE WE RUNNING UNDER XXDP/ACT?
2008 002432 001006 BNE 67$ ;BRANCH IF YES
2009 002434 023727 001140 000176 CMP $SWR,$SWREG ;SOFTWARE SWITCH REG SELECTED?
2010 002442 001005 BNE 68$ ;BRANCH IF NO
2011 002444 104406 GTSWR ;GET SOFT-SWR SETTINGS
2012 002446 000403 BR 68$
2013 002450 112737 000001 001134 67$: MOVB #1,$AUTOB ;SET AUTO-MODE INDICATOR

```

```

2014 002456
2015 002456 004737 003050
2016 002462 104401 071437
2017 002466 104412
2018 002470 012637 001224
2019 002474 023727 001224 000007
2020 002502 101403
2021 002504 104401 071467
2022 002510 000744
2023 002512 013737 001224 001226 2$:
2024 002520 005237 001226
2025 002524 042737 000016 001226
2026 002532 013746 001224
2027 002536 042716 177771
2028 002542 052637 001226
2029 002546 104401 071511
2030 002552 013746 001224
2031
2032 002556 104403
2033 002560 001
2034 002561 000
2035 002562 104401 071537
2036 002566 013746 001226
2037
2038 002572 104403
2039 002574 001
2040 002575 000
2041 002576 104401 001207
2042 002602 013737 001224 001230
2043 002610 062737 000006 001230
2044 002616 042737 177770 001230
2045 002624 013701 001224
2046 002630 116137 101002 001236
2047 002636 005037 001262
2048 002642 005037 001264
2049 002646 005037 001270
2050 002652 005037 001272
2051 002656 004737 065766
2052 002662 000137 002676
2053 002666 104401 071565
2054 002672 000000
2055 002674 000776
2056
2057
2058
2059 002676 000005
2060 002700 005037 177776
2061 002704 104401 001207
2062 002710 013700 001304
2063 002714 012706 001100
2064 002720 004737 065766
2065 002724 000240
2066 002726 004737 067664
2067 002732 005037 001300
2068 002736 005037 001100
2069 002742 112737 000001 001115

68$:
JSR PC,CHANGE ;CHECK/CHANGE THE RH11 ADDRESS
TYPE ,ENTERA ;ENTER DRIVE ADDRESS
RDOCT ;GET THE ADDRESS
MOV (SP)+,PORTA ;STORE THE ADDRESS
CMP PORTA,#7 ;SEE IF ADDRESS TOO LARGE
BLOS 2$ ;BR IF NOT
TYPE ADRERR ;TYPE ADDRESS ERROR MESSAGE
BR 1$ ;TRY AGAIN
MOV PORTA,PORTB ;GENERATE THE PORT B ADDRESS
INC PORTB ;INCREMENT THE ADDRESS
BIC #16,PORTB ;LEAVE BIT 0
MOV PORTA,-(SP) ;PUT PORT A ADDRESS ON THE STACK
BIC #16,(SP) ;SAVE BITS 1 & 2
BIS (SP)+,PORTB ;SET BITS 1 & 2 IN PORT B ADDRESS
TYPE PORTAIS ;'PORT A ADDRESS IS '
MOV PORTA,-(SP) ;SAVE PORTA FOR TYPEOUT
TYPE PORTA ADDRESS ;TYPE PORT A ADDRESS
GO TYPE--OCTAL ASCII ;GO TYPE--OCTAL ASCII
TYPE 1 DIGIT(S) ;TYPE 1 DIGIT(S)
SUPPRESS LEADING ZEROS ;SUPPRESS LEADING ZEROS
TYPE PORTBIS ;'PORT B ADDRESS IS '
MOV PORTB,-(SP) ;SAVE PORTB FOR TYPEOUT
TYPE PORTB ADDRESS ;TYPE PORT B ADDRESS
GO TYPE--OCTAL ASCII ;GO TYPE--OCTAL ASCII
TYPE 1 DIGIT(S) ;TYPE 1 DIGIT(S)
SUPPRESS LEADING ZEROS ;SUPPRESS LEADING ZEROS
TYPE $CRLF ;ANOTHER CR-LF
MOV PORTA,PORTC ;GENERATE ADDRESS OF DRIVE NOT TESTED
ADD #6,PORTC ;COMPLEMENT SOME BITS
BIC #17,PORTC ;SAVE ONLY LOWER BITS
MOV PORTA,R1 ;USE PORT A ADDRESS AS INDEX
MOVB ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
CLR TIMEA ;CLEAR TIMEOUT ONE--SHOT VALUE LOCATION
CLR TIMEAP ;CLEAR TIMEOUT ONE--SHOT VALUE LOCATION
CLR TIMEB ;CLEAR TIMEOUT ONE--SHOT VALUE LOCATION
CLR TIMEBP ;CLEAR TIMEOUT ONE--SHOT VALUE LOCATION
JSR PC,CKCLK ;SETUP CLOCK
JMP EXEC ;CLOCK HAS BEEN STARTED
TYPE ,NOCLOCK ;NO CLOCK ON SYSTEM
HALT ;FATAL ERROR
BR 3$ ;INTERLOCK THE HALT

3$:

;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
EXEC: RESET
CLR PS ;CLEAR EVERYTHING
TYPE $CRLF ;CLEAR THE PROCESSOR STATUS WORD
MOV $RMADR,R0 ;CR-LF
MOV $STACK,SP ;RH11 ADDRESS FOR INDEXING
JSR PC,CKCLK ;LOAD STACK POINTER
NOP ;START THE CLOCK
JSR PC,$TKINT ;RETURN IF NO CLOCK
CLR KYBCTL ;INITIALIZE THE KEYBOARD
CLR $PASS ;CLEAR SINGLE TEST INDICATOR
MOVB #1,$ERMAX ;CLEAR THE PASS COUNT
;SET ERROR MAX TO 1

```



```

2070 002750 012737 002750 001106      MOV      #.,$LPADR      ;INITIAL SETTING FOR LOOP ADDRESS
2071 002756 012737 002756 001110      MOV      #.,$LPERP     ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
2072 002764 104401 071634      1$:     TYPE      ,TESTNO ;ASK FOR TEST NUMBER
2073 002770 104412                RDOCT                ;GET THE NUMBER
2074 002772 012601                MOV      (SP)+,R1     ;PUT ENTRY INTO R1
2075 002774 001002                BNE      2$          ;BR IF NOT ZERO
2076 002776 000137 003164      JMP      TST1        ;ENTER ZERO - PERFORM ALL TESTS
2077 003002 020137 101012      2$:     CMP      R1,MAXTN ;SEE IF NUMBER GREATER THAN MAXIMUM
2078 003006 003403                BLE      3$          ;BR IF LESS OR EQUAL
2079 003010 104401 071654      TYPE      ,BADNO     ;BAD ENTRY
2080 003014 000763                BR       1$          ;TRY AGAIN
2081 003016 005301                3$:     DEC      R1     ;DECREMENT ENTRY
2082 003020 006301                ASL      R1          ;SHIFT IT LEFT
2083 003022 016137 100666 003046      MOV      TSTADR(R1),4$ ;GET THE TEST ADDRESS
2084 003030 005237 001300      INC      KYBCTL      ;SET SINGLE TEST INDICATOR
2085 003034 012737 000001 001104      MOV      #1,$ICNT    ;PRESET ITERATION COUNT
2086 003042 000177 000000      JMP      24$         ;GO TO THE SELECTED TEST
2087 003046 000000      4$:     .WORD      0    ;TEST ADDRESS GOES HERE
2088
2089      ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
2090
2091 003050 005737 001302      CHANGE: TST      CHGADR ;CHANGE THE ADDRESS ?
2092 003054 001421                BEQ      3$          ;BR IF NOT
2093 003056 005037 001302      CLR      CHGADR     ;CLEAR THE INDICATOR
2094 003062 104401 071714      1$:     TYPE      ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS
2095 003066 013746 001304      MOV      $RMADR,-(SP) ;PUT THE ADDRESS ON THE STACK
2096 003072 104402                TYPOC                ;TYPE THE ACTUAL ADDRESS
2097 003074 104401 001207      TYPE      ,$CRLF     ;CR-LF
2098 003100 104401 071774      TYPE      ,NTRH11    ;ASK FOR NEW ADDRESS
2099 003104 104412                RDOCT
2100 003106 005716                TST      (SP)        ;0 OR 'CR' ENTERED ?
2101 003110 001402                BEQ      2$          ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
2102 003112 011637 001304      MOV      (SP), $RMADR ;NEW RH11 ADDRESS
2103 003116 005726                2$:     TST      (SP)+  ;CORRECT THE STACK POINTER
2104 003120 012737 003140 000004      3$:     MOV      #4,$2#4  ;LOAD TRAP ADDRESS
2105 003126 013700 001304      MOV      $RMADR,RO   ;RH11 ADDRESS
2106 003132 005760 000002      TST      RMWC(RO)    ;SEE IF RH11 RESPONDS AT THAT ADDRESS
2107 003136 000404                BR       5$          ;BR, RH11 ALIVE AT PRESENT ADDRESS
2108 003140 104056                4$:     ERROR      56   ;NO RESPONSE TO ADDRESS
2109 003142 062706 000004      ADD      #4,SP       ;RESET THE STACK POINTER
2110 003146 000745                BR       1$          ;GET ADDRESS AGAIN
2111 003150 012737 000006 000004      5$:     MOV      #6,$2#4 ;RESTORE THE VECTOR
2112 003156 000207                RTS      PC          ;RETURN
2113
2114      ;;*****
2115
2116      .SBTTL  *** TESTS ***
2117
2118      ;;*****
2119
2120
2121 003160 013700 001304      TST1AA: MOV      $RMADR,RO ;;RESTORE RO AFTER END OF PASS
2122
2123      ;;*****
2124      ;#TEST 1      NEUTRAL ACCESS TEST
2125      ;*

```

```

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

```

```

; *VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS
; *
; * A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
; * DRIVE IS A DUAL PORT RM03, THAT THE DRIVE IS ONLINE (RMDS1 HAS
; * 'MOL' 'PGM' 'DPR' & 'DAY' BITS SET), AND THE THE DRIVE SERIAL
; * NUMBER READ THROUGH BOTH PORTS IS THE SAME.
; *
; * B. THE TEST IS REPEATED THROUGH BOTH PORTS.
; *
; *****
; *ST1:
; 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 #1, $STNM ; TEST NUMBER
; MOV #TEST1, $LPADR ; LOAD LOOP ON TEST ADDRESS
; MOV #TEST1, $LPERR ; LOAD LOOP ON ERROR ADDRESS
; MOV #1, $TIMES ; DO 1 ITERATION
TEST1: MOV #STACK, SP ; LOAD THE STACK POINTER
; MOV #CLR, RMCS2(R0) ; INITIALIZE THE MASSBUS
; *****
; *VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B
; *****
; MOVB PORTA, RMCS2(R0) ; SELECT PORT A
; MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
; TST RMDS1(R0) ; SEE IF DRIVE (PORT A) PRESENT
; CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
; MOV RMCS2(R0), $BDDAT ; GET CONTENTS OF RMCS2
; MOV #RMCS2, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
; ADD R0, $B0ADR ; ADD RH11 BASE ADDRESS
; CLR $GDDAT ; WHAT REGISTER SHOULD BE
; MOV $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
; BIC #CNED, $TMP0 ; SAVE SPECIFIED BITS
; CMP $GDDAT, $TMP0 ; COMPARE THE BITS
; BEQ 64$ ; BR IF OK
; MOV $BDDAT, $TMP4 ; COPY 'BAD DATA'
; BIC #NED, $TMP4 ; CLEAR THE MASKED BITS
; BIS $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
; ERROR 37 ; TYPE MESSAGE 37
; COM CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
64$: NOP
; TST CKERR ; WAS 'NED' SET ?
; BEQ .+10 ; BR IF NOT
; MOV #CLR, RMCS2(R0) ; ISSUE MASSBUS INIT TO CLEAR 'NED'
; MOVB PORTB, RMCS2(R0) ; SELECT PORT B
; MOV PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
; TST RMDS1(R0) ; SEE IF DRIVE (PORT B) PRESENT
; CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
; MOV RMCS2(R0), $BDDAT ; GET CONTENTS OF RMCS2
; MOV #RMCS2, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
; ADD R0, $B0ADR ; ADD RH11 BASE ADDRESS
; CLR $GDDAT ; WHAT REGISTER SHOULD BE
; MOV $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'

```

```

2182 003470 042737 167777 001164 BIC #1CNE0,$STMP0 ;SAVE SPECIFIED BITS
2183 003476 023737 001124 001164 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
2184 003504 001414 BEQ 66$ ;BR IF OK
2185 003506 013737 001126 001174 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
2186 003514 042737 010000 001174 BIC #NED,$STMP4 ;CLEAR THE MASKED BITS
2187 003522 053737 001174 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2188 003530 104037 ERROR 37 ;TYPE MESSAGE 37
2189 003532 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2190 003536 000240 66$: NOP
2191 003540 005737 001250 TST CKERR ;WAS 'NED' SET ?
2192 003544 001403 BEQ .+10 ;BR IF NOT
2193 003546 012760 000040 000010 MOV #CLR,RMCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2194
2195 ;:*****
2196 ;CONFIRM THAT DRIVE IS AN RM03 AND IS DUAL PORT
2197
2198 003554 113760 001224 000010 MOV#B PORTA,RMCS2(RO) ;SELECT PORT A
2199 003562 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2200 003570 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2201 003574 016037 000026 001126 MOV RMDT(RO),$BDDAT ;GET CONTENTS OF RMDT
2202 003602 012737 000026 001122 MOV #RMDT,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2203 003610 060037 001122 ADD RO,$B0ADR ;ADD R#11 BASE ADDRESS
2204 003614 012737 024024 001124 MOV #024024,$GDDAT ;WHAT REGISTER SHOULD BE
2205 003622 022737 024025 001126 CMP #24025,$BDDAT ;DUAL PORT RM02 ?
2206 003630 001426 BEQ 68$ ;BRANCH IF SO
2207 003632 013737 001126 001164 MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
2208 003640 042737 000003 001164 BIC #1C177774,$STMP0 ;SAVE SPECIFIED BITS
2209 003646 023737 001124 001164 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
2210 003654 001414 BEQ 68$ ;BR IF OK
2211 003656 013737 001126 001174 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
2212 003664 042737 177774 001174 BIC #177774,$STMP4 ;CLEAR THE MASKED BITS
2213 003672 053737 001174 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2214 003700 104001 ERROR 1 ;TYPE MESSAGE 1
2215 003702 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2216 003706 000240 68$: NOP
2217 003710 113760 001226 000010 MOV#B PORTB,RMCS2(RO) ;SELECT PORT B
2218 003716 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2219 003724 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2220 003730 016037 000026 001126 MOV RMDT(RO),$BDDAT ;GET CONTENTS OF RMDT
2221 003736 012737 000026 001122 MOV #RMDT,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2222 003744 060037 001122 ADD RO,$B0ADR ;ADD R#11 BASE ADDRESS
2223 003750 012737 024024 001124 MOV #024024,$GDDAT ;WHAT REGISTER SHOULD BE
2224 003756 022737 024025 001126 CMP #24025,$BDDAT ;DUAL PORT RM02 ?
2225 003764 001426 BEQ 70$ ;BRANCH IF SO
2226 003766 013737 001126 001164 MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
2227 003774 042737 000003 001164 BIC #1C177774,$STMP0 ;SAVE SPECIFIED BITS
2228 004002 023737 001124 001164 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
2229 004010 001414 BEQ 70$ ;BR IF OK
2230 004012 013737 001126 001174 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
2231 004020 042737 177774 001174 BIC #177774,$STMP4 ;CLEAR THE MASKED BITS
2232 004026 053737 001174 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2233 004034 104001 ERROR 1 ;TYPE MESSAGE 1
2234 004036 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2235 004042 000240 70$: NOP
2236
2237 ;:*****

```

```

;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
2238
2239
2240 004044 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
2241 004052 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2242 004060 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2243 004064 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
2244 004072 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2245 004100 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2246 004104 012737 001000 001124 MOV #PGM, $GDDAT ;WHAT REGISTER SHOULD BE
2247 004112 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2248 004120 042737 176777 001164 BIC #1CPGM, $TMP0 ;SAVE SPECIFIED BITS
2249 004126 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
2250 004134 001414 BEQ 72$ ;BR IF OK
2251 004136 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2252 004144 042737 001000 001174 BIC #PGM, $TMP4 ;CLEAR THE MASKED BITS
2253 004152 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2254 004160 104045 ERROR 45 ;TYPE MESSAGE 45
2255 004162 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2256 004166 000240 72$: NOP
2257 004170 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2258 004174 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
2259 004202 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2260 004210 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2261 004214 012737 010600 001124 MOV #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
2262 004222 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2263 004230 042737 167177 001164 BIC #1C10600, $TMP0 ;SAVE SPECIFIED BITS
2264 004236 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
2265 004244 001414 BEQ 74$ ;BR IF OK
2266 004246 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2267 004254 042737 010600 001174 BIC #10600, $TMP4 ;CLEAR THE MASKED BITS
2268 004262 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2269 004270 104002 ERROR 2 ;TYPE MESSAGE 2
2270 004272 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2271 004276 000240 74$: NOP
2272 004300 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
2273 004306 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2274 004314 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2275 004320 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
2276 004326 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2277 004334 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2278 004340 012737 001000 001124 MOV #PGM, $GDDAT ;WHAT REGISTER SHOULD BE
2279 004346 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2280 004354 042737 176777 001164 BIC #1CPGM, $TMP0 ;SAVE SPECIFIED BITS
2281 004362 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
2282 004370 001414 BEQ 76$ ;BR IF OK
2283 004372 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2284 004400 042737 001000 001174 BIC #PGM, $TMP4 ;CLEAR THE MASKED BITS
2285 004406 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2286 004414 104045 ERROR 45 ;TYPE MESSAGE 45
2287 004416 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2288 004422 000240 76$: NOP
2289 004424 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2290 004430 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
2291 004436 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2292 004444 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2293 004450 012737 010600 001124 MOV #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE

```

```

2294 004456 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2295 004464 042737 167177 001164 BIC #1C10600,$TMP0 ;SAVE SPECIFIED BITS
2296 004472 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
2297 004500 001414 78$ BEQ 78$ ;BR IF OK
2298 004502 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
2299 004510 042737 010600 001174 BIC #10600,$TMP4 ;CLEAR THE MASKED BITS
2300 004516 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2301 004524 104202 ERROR 2 ;TYPE MESSAGE 2
2302 004526 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2303 004532 000240 78$: NOP

```

```

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

```

```

2308 004534 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
2309 004542 016037 000030 001124 MOV RMSN(RO), $GDDAT ;STORE THE PORT A SERIAL NUMBER
2310 004550 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
2311 004556 016037 000030 001126 MOV RMSN(RO), $BDDAT ;STORE THE PORT B SERIAL NUMBER
2312 004564 023737 001124 001126 CMP $GDDAT,$BDDAT ;ARE THEY THE SAME ?
2313 004572 001406 BEQ 1$ ;BR IF THEY ARE
2314 004574 104003 ERROR 3 ;REPORT THE ERROR
2315 004576 032777 100000 174334 BIT #SW15,$SWR ;HALT ON ERROR ?
2316 004604 001001 BNE 1$ ;BR IF SET - PROGRAM HAS ALREADY HALTED
2317 004606 000000 HALT ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
2318 004610 000004 1$: SCOPE ;LOOP ?

```

```

*****
*TEST 2 PORT 'A' SEIZE/TIMEOUT TEST
*
*VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT
* IT CAN BE RELEASED BY THE ONE SECOND TIMER.
*
* A. WRITE 0'S INTO RMDA THROUGH PORT 'A'; VERIFY THAT THE DRIVE
* HAS BEEN SEIZED.
*
* B. READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B';
* VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
*
* C. WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
* MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
* VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
* NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.
*
*****

```

```

2339 004612 005737 001300 1$T2: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2340 004612 001406 BEQ 2$ ;BR IF NOT
2341 004616 001406 BPL 1$ ;BR IF JUST ENTERED TEST
2342 004620 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2343 004622 000137 002676 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2344 004626 012737 177777 001300 2$: MOVB #2,$STNM ;TEST NUMBER
2345 004634 112737 000002 001102 MOV #TEST2,$LPADR ;LOAD LOOP ON TEST ADDRESS
2346 004642 012737 004664 001106 MOV #TEST2,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2347 004650 012737 004664 001110 MOV #2,$TIMES ;DO 2. ITERATIONS
2348 004656 012737 000002 001176 TEST2: MOV #STACK,$SP ;LOAD THE STACK POINTER
2349 004664 012706 001100

```

```

2350 004670 012737 000240 177776      MOV      #(<5*32.),2#PS ;SET PRIORITY TO 5 IN CASE LOOPING
2351 004676 005037 001262      CLR      TIMEA          ;CLEAR TIMEOUT VALUE FOR PORT A
2352 004702 005037 001264      CLR      TIMEAP        ;CLEAR UPPER TIMEOUT TOLERANCE
2353 004706 005037 001266      CLR      TIMEAM        ;CLEAR LOWER TIMEOUT TOLERANCE
2354
2355      ;*****
2356      ;START THE TIMER
2357
2358 004712 005037 001256      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
2359 004716 012737 003720 001260      MOV      #2000.,WATCH ;SET WATCH TO 2000 MS
2360
2361      ;*****
2362
2363      ;SEIZE THE DRIVE THROUGH PORT A
2364
2365 004724 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
2366 004732 013737 001224 001242      MOV      PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
2367 004740 005060 000006      CLR      RMDA(RO)      ;WRITE RMDA
2368 004744 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
2369 004752 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2370 004760 013737 001226 001244      MOV      PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
2371 004766 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
2372 004774 010037 001122      MOV      RO, $BDDADR ;R#11 BASE ADDRESS
2373 005000 062737 000012 001122      ADD      #RMD51, $BDDADR ;GENERATE BAD REGISTER ADDRESS
2374 005006 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
2375 005012 023737 001124 001126      CMP      $GDDAT, $BDDAT ;IS THE REGISTER ZERO
2376 005020 001403 64$          BEQ      64$          ;BR IF IT IS
2377 005022 104004      ERROR   4            ;REPORT THE ERROR
2378 005024 000137 006156      JMP      5$          ;BYPASS REST OF THE SUBTEST
2379 005030
2380 005030 113760 001224 000010      64$: MOV      PORTA, RMCS2(RO) ;SELECT PORT A
2381 005036 013737 001224 001240      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2382 005044 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2383 005052 042737 020001 001126      BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
2384 005060 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY, $GDDAT ;EXPECTED STATUS
2385 005066 013737 001124 001166      MOV      $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
2386 005074 005137 001166      COM      $TMP1        ;COMPLEMENT THE EXPECTED STATUS
2387 005100 013737 001126 001164      MOV      $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
2388 005106 043737 001166 001164      BIC      $TMP1, $TMP0 ;CLEAR UNWANTED BITS
2389 005114 023737 001124 001164      CMP      $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
2390 005122 001401 65$          BEQ      65$          ;BR IF THEY ARE
2391 005124 104005      ERROR   5            ;REPORT THE ERROR
2392 005126 000240      65$: NOP
2393
2394      ;*****
2395      ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2396
2397 005130 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
2398 005136 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2399 005144 016046 000046      MOV      RMEC2(RO), -(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
2400 005150 016046 000044      MOV      RMEC1(RO), -(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
2401 005154 016046 000030      MOV      RMSN(RO), -(SP) ;STORE REGISTER RMSN, PORT B, FOR CHECK
2402 005160 016046 000034      MOV      RMDC(RO), -(SP) ;STORE REGISTER RMDC, PORT B, FOR CHECK
2403 005164 016046 000032      MOV      RMOF(RO), -(SP) ;STORE REGISTER RMOF, PORT B, FOR CHECK
2404 005170 016046 000042      MOV      RMER2(RO), -(SP) ;STORE REGISTER RMER2, PORT B, FOR CHECK
2405 005174 016046 000020      MOV      RMLA(RO), -(SP) ;STORE REGISTER RMLA, PORT B, FOR CHECK

```

```

2406 005200 016046 000026      MOV      RMDT(RO),-(SP)  ;STORE REGISTER RMDT, PORT B, FOR CHECK
2407 005204 016046 000006      MOV      RMDA(RO),-(SP)  ;STORE REGISTER RMDA, PORT B, FOR CHECK
2408 005210 016046 000024      MOV      RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT B, FOR CHECK
2409 005214 016046 000014      MOV      RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT B, FOR CHECK
2410
2411      ;*****
2412      ;WAIT FOR PORT A TO TIMEOUT
2413
2414 005220 005760 000012      1$:     TST      RMD51(RO)  ;WAIT FOR THE DRIVE TO TIMEOUT
2415 005224 001006                BNE      2$              ;BR WHEN TIMEOUT OCCURS
2416 005226 005737 001260      TST      WATCH          ;CHECK WATCH
2417 005232 001372                BNE      1$              ;BR IF NOT ZERO
2418 005234 104036                ERROR    36              ;NO TIMEOUT WITHIN 2 SECONDS
2419 005236 000137 005642      JMP      4$              ;BYPASS TIMEOUT TIME CHECK
2420 005242 012737 000340      2$:     MOV      #(<7*32.>),D#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2421 005250 013737 001256      MOV      TIME,TIMEA      ;SAVE THE ELAPSED TIME FOR PORT A
2422 005256 004537 066162      JSR      RS,TOLER        ;CALCULATE THE TOLERANCE
2423 005262 001262                .WORD    TIMEA           ;TIMEOUT VALUE FOR PORT A
2424 005264 012637 001264      MOV      (SP)+,TIMEAP     ;+25% TOLERANCE
2425 005270 012637 001266      MOV      (SP)+,TIMEAM     ;-25% TOLERANCE
2426
2427      ;*****
2428      ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2429
2430 005274 023727 001256 000764      CMP      TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
2431 005302 103001                BHS      3$              ;BR IF IT WAS
2432 005304 104055                ERROR    55              ;REPORT TIMEOUT TOO SHORT
2433
2434      ;*****
2435      ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2436
2437 005306 012737 000240 177776      3$:     MOV      #(<5*32.>),D#PS ;RESTORE PRIORITY TO 5
2438
2439      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2440
2441 005314 005037 001254                CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
2442 005320 012737 000012 001122      MOV      #RMD51,$BDADR   ;FORM THE ADDRESS OF RMD51 FOR TYPEOUT
2443 005326 060037 001122      ADD      RO,$BDADR        ;ADD THE I/O BASE ADDRESS
2444 005332 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2445 005340 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
2446 005346 016037 000012 001170      MOV      RMD51(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2447 005354 042737 024001 001170      BIC      #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
2448 005362 013737 001170 001164      MOV      STMP2,STMP0      ;COPY IT INTO 'STMP0'
2449 005370 042737 100100 001164      BIC      #ATA!VV,STMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2450 005376 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
2451 005404 016037 000012 001172      MOV      RMD51(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2452 005412 042737 024001 001172      BIC      #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
2453 005420 013737 001172 001166      MOV      STMP3,STMP1      ;COPY IT INTO 'STMP1'
2454 005426 042737 100100 001166      BIC      #ATA!VV,STMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2455 005434 023737 001164 001166      CMP      STMP0,STMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2456 005442 001006                BNE      66$             ;BR IF NOT
2457 005444 005737 001164      TST      STMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2458 005450 001037                BNE      68$             ;BR IF NOT
2459 005452 104046                ERROR    46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2460 005454 000137 005640      JMP      70$             ;BYPASS THE REST OF THE CHECKS
2461 005460 013737 001170 001126 66$:     MOV      STMP2,$BDADR     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE

```


K04

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 49
T2 PORT 'A' SEIZE/TIMEOUT TEST

SEQ 0049

2462	005466	013737	001226	001240		MOV	PORTB,PTNBR	SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2463	005474	113760	001226	000010		MOV	PORTB, RMCS2(RO)	SELECT PORT B.
2464	005502	005737	001164			TST	\$TMP0	SEE IF STATUS EQ 0 FROM PORT A.
2465	005506	001414				BEQ	67\$	BR IF ZERO
2466	005510	013737	001224	001240		MOV	PORTA,PTNBR	SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2467	005516	013737	001172	001126		MOV	\$TMP3,\$BDDAT	'BAD DATA' FOR ERROR TYPE OUT
2468	005524	113760	001224	000010		MOV	PORTA, RMCS2(RO)	SELECT PORT A.
2469	005532	005737	001166			TST	\$TMP1	SEE IF STATUS EQ ZERO FROM PORT B.
2470	005536	001004				BNE	68\$	BR IF NOT
2471	005540	012737	177777	001254	67\$:	MOV	#-1,RELEERR	SET 'RELEASE ERROR' INDICATOR
2472	005546	104022				ERROR	22	TYPE ERROR MESSAGE 22
2473	005550	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	LOOK FOR BIT FAILURES WHEN RMD51 READ
2474	005556	013737	001224	001240		MOV	PORTA,PTNBR	CHANGE PORT NUMBER
2475	005564	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	DON'T CHECK ATTN BIT OR VV BIT
2476	005572	023737	001124	001126		CMF	\$GDDAT,\$BDDAT	ALL BITS OK ?
2477	005600	001401				BEQ	69\$	BR IF OK FROM PORT A.
2478	005602	104007				ERROR	7	REPORT ERROR
2479	005604	013737	001172	001126	69\$:	MOV	\$TMP3,\$BDDAT	CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
2480	005612	013737	001226	001240		MOV	PORTB,PTNBR	CHANGE PORT NUMBER
2481	005620	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	DON'T CHECK ATTN BIT OR VV BIT
2482	005626	023737	001124	001126		CMF	\$GDDAT,\$BDDAT	SEE IF READ OK FROM PORT B.
2483	005634	001401				BEQ	70\$	BR IF OK
2484	005636	104007				ERROR	7	REPORT ERROR
2485	005640	000240			70\$:	NOP		

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

2491	005642	013737	001226	001240	4\$:	MOV	PORTB,PTNBR	CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2492	005650	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMER1
2493	005654	062737	000014	001122		ADD	#RMER1,\$B0ADR	ADDRESS OF RMER1 FOR TIMEOUT
2494	005662	012637	001126			MOV	(SP)+,\$BDDAT	CHECK THE STORED CONTENTS OF RMER1
2495	005666	001401				BEQ	.+4	CONTENTS ZERO ?
2496	005670	104006				ERROR	6	REPORT THAT PORT B SAW NON-ZERO REGISTER
2497	005672	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMMR1
2498	005676	062737	000024	001122		ADD	#RMMR1,\$B0ADR	ADDRESS OF RMMR1 FOR TIMEOUT
2499	005704	012637	001126			MOV	(SP)+,\$BDDAT	CHECK THE STORED CONTENTS OF RMMR1
2500	005710	001401				BEQ	.+4	CONTENTS ZERO ?
2501	005712	104006				ERROR	6	REPORT THAT PORT B SAW NON-ZERO REGISTER
2502	005714	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMDA
2503	005720	062737	000006	001122		ADD	#RMDA,\$B0ADR	ADDRESS OF RMDA FOR TIMEOUT
2504	005726	012637	001126			MOV	(SP)+,\$BDDAT	CHECK THE STORED CONTENTS OF RMDA
2505	005732	001401				BEQ	.+4	CONTENTS ZERO ?
2506	005734	104006				ERROR	6	REPORT THAT PORT B SAW NON-ZERO REGISTER
2507	005736	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMDT
2508	005742	062737	000026	001122		ADD	#RMDT,\$B0ADR	ADDRESS OF RMDT FOR TIMEOUT
2509	005750	012637	001126			MOV	(SP)+,\$BDDAT	CHECK THE STORED CONTENTS OF RMDT
2510	005754	001401				BEQ	.+4	CONTENTS ZERO ?
2511	005756	104006				ERROR	6	REPORT THAT PORT B SAW NON-ZERO REGISTER
2512	005760	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMLA
2513	005764	062737	000020	001122		ADD	#RMLA,\$B0ADR	ADDRESS OF RMLA FOR TIMEOUT
2514	005772	012637	001126			MOV	(SP)+,\$BDDAT	CHECK THE STORED CONTENTS OF RMLA
2515	005776	001401				BEQ	.+4	CONTENTS ZERO ?
2516	006000	104006				ERROR	6	REPORT THAT PORT B SAW NON-ZERO REGISTER
2517	006002	010037	001122			MOV	RO,\$B0ADR	BASE ADDRESS FOR REGISTER RMER2

```

2518 006006 062737 000042 001122 ADD #RMR2,$BDADR ;ADDRESS OF RMR2 FOR TYPEOUT
2519 006014 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMR2
2520 006020 001401 BEQ .+4 ;CONTENTS ZERO ?
2521 006022 104006 ERROR 6 ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2522 006024 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMOF
2523 006030 062737 000032 001122 ADD #RMOF,$BDADR ;ADDRESS OF RMOF FOR TYPEOUT
2524 006036 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMOF
2525 006042 001401 BEQ .+4 ;CONTENTS ZERO ?
2526 006044 104006 ERROR 6 ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2527 006046 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDC
2528 006052 062737 000034 001122 ADD #RMDC,$BDADR ;ADDRESS OF RMDC FOR TYPEOUT
2529 006060 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMDC
2530 006064 001401 BEQ .+4 ;CONTENTS ZERO ?
2531 006066 104006 ERROR 6 ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2532 006070 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMSN
2533 006074 062737 000030 001122 ADD #RMSN,$BDADR ;ADDRESS OF RMSN FOR TYPEOUT
2534 006102 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMSN
2535 006106 001401 BEQ .+4 ;CONTENTS ZERO ?
2536 006110 104006 ERROR 6 ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2537 006112 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC1
2538 006116 062737 000044 001122 ADD #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TYPEOUT
2539 006124 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMEC1
2540 006130 001401 BEQ .+4 ;CONTENTS ZERO ?
2541 006132 104006 ERROR 6 ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2542 006134 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC2
2543 006140 062737 000046 001122 ADD #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TYPEOUT
2544 006146 012637 001126 MOV (SP)+,$BDADR ;CHECK THE STORED CONTENTS OF RMEC2
2545 006152 001401 BEQ .+4 ;CONTENTS ZERO ?
2546 006154 104006 ERROR 6 ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2547 006156 000004 5$: SCOPE LOOP ?

```

```

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

```

```

2567 006160 005737 001300 †ST3: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ^
2568 006160 001406 BEQ 2$ ;BR IF NOT
2569 006164 100002 BPL 1$ ;BR IF JUST ENTERED TEST
2570 006166 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2571 006170 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2572 006174 112737 000003 001102 2$: MOVB #3,$TSTNM ;TEST NUMBER
2573

```

MO4

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 51
T3 PORT 'B' SEIZE/TIMEOUT TEST

SEQ 0051

```

2574 006210 012737 006232 001106      MOV      #TEST3,$LPADR ;LOAD LOOP ON TEST ADDRESS
2575 006216 012737 006232 001110      MOV      #TEST3,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2576 006224 012737 000002 001176      MOV      #2,$TIMES ;DO 2 ITERATIONS
2577 006232 012706 001100      TEST3: MOV    #STACK,SP ;LOAD THE STACK POINTER
2578 006236 012737 000240 177776      MOV    #(<5*32.>),@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
2579 006244 005037 001270      CLR     TIMEB ;CLEAR TIMEOUT VALUE FOR PORT B
2580 006250 005037 001272      CLR     TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
2581 006254 005037 001274      CLR     TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE
2582
2583 ;*****
2584 ;START THE TIMER
2585
2586 006260 005037 001256      CLR     TIME ;CLEAR THE ELAPSED TIME COUNTER
2587 006264 012737 003720 001260      MOV    #2000.,WATCH ;SET WATCH TO 2000 MS
2588
2589 ;*****
2590
2591 ;SEIZE THE DRIVE THROUGH PORT B
2592
2593 006272 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
2594 006300 013737 001226 001242      MOV    PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
2595 006306 005060 000006      CLR    RMDA(RO) ;WRITE RMDA
2596 006312 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
2597 006320 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2598 006326 013737 001224 001244      MOV    PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
2599 006334 016037 000012 001126      MOV    RMSD1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
2600 006342 010037 001122      RO, $BDAOR ;RHLI BASE ADDRESS
2601 006346 062737 000012 001122      ADD    #RMSD1, $BDAOR ;GENERATE BAD REGISTER ADDRESS
2602 006354 005037 001124      CLR    $GDDAT ;REGISTER SHOULD BE ZERO
2603 006360 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER ZERO
2604 006366 001403      BEQ    64$ ;BR IF IT IS
2605 006370 104004      ERROR 4 ;REPORT THE ERROR
2606 006372 000137 007524      JMP    5$ ;BYPASS REST OF THE SUBTEST
2607 006376
2608 006376 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
2609 006404 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2610 006412 016037 000012 001126      MOV    RMSD1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2611 006420 042737 020001 001126      BIC    #OM!PIP,$GDDAT ;CLEAR DONT CARE BITS
2612 006426 012737 011600 001124      MOV    #MOL!PGA!DPR!DRY,$GDDAT ;EXPECTED STATUS
2613 006434 013737 001124 001166      MOV    $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
2614 006442 005137 001166      COM    $TMP1 ;COMPLEMENT THE EXPECTED STATUS
2615 006446 013737 001126 001164      MOV    $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
2616 006454 043737 001166 001164      BIC    $TMP1, $TMP0 ;CLEAR UNWANTED BITS
2617 006462 023737 001124 001164      CMP    $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
2618 006470 001401      BEQ    65$ ;BR IF THEY ARE
2619 006472 104005      ERROR 5 ;REPORT THE ERROR
2620 006474 000240      NOP
2621
2622 ;*****
2623 ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2624
2625 006476 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
2626 006504 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2627 006512 016046 000046      MOV    RMEC2(RO), -(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
2628 006516 016046 000044      MOV    RMEC1(RO), -(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
2629 006522 016046 000030      MOV    RMSN(RO), -(SP) ;STORE REGISTER RMSN, PORT A, FOR CHECK

```

```

2630 006526 016046 000034      MOV      RMDC(RO),-(SP) ;STORE REGISTER RMDC, PORT A, FOR CHECK
2631 006532 016046 000032      MOV      RMOF(RO),-(SP) ;STORE REGISTER RMOF, PORT A, FOR CHECK
2632 006536 016046 000042      MOV      RMER2(RO),-(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
2633 006542 016046 000020      MOV      RMLA(RO),-(SP) ;STORE REGISTER RMLA, PORT A, FOR CHECK
2634 006546 016046 000026      MOV      RMDT(RO),-(SP) ;STORE REGISTER RMDT, PORT A, FOR CHECK
2635 006552 016046 000006      MOV      RMDA(RO),-(SP) ;STORE REGISTER RMDA, PORT A, FOR CHECK
2636 006556 016046 000024      MOV      RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
2637 006562 016046 000014      MOV      RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK
2638
2639
2640      ;*****
2641      ;WAIT FOR PORT B TO TIMEOUT
2642 006566 005760 000012      1S:    TST      RMDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
2643 006572 001006                BNE      2S          ;BR WHEN TIMEOUT OCCURS
2644 006574 005737 001260      TST      WATCH      ;CHECK WATCH
2645 006600 001372                BNE      1S          ;BR IF NOT ZERO
2646 006602 104036                ERROR    36          ;NO TIMEOUT WITHIN 2 SECONDS
2647 006604 000137 007210      JMP      4S          ;BYPASS TIMEOUT TIME CHECK
2648 006610 012737 000340      177776 2S:    MOV      #<7*32.>,2#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2649 006616 013737 001256      001270      MOV      TIME,TIMEB  ;SAVE THE ELAPSED TIME FOR PORT B
2650 006624 004537 066162      JSR      RS,TOLER  ;CALCULATE THE TOLERANCE
2651 006630 001270                .WORD    TIMEB      ;TIMEOUT VALUE FOR PORT B
2652 006632 012637 001272      MOV      (SP)+,TIMEBP ;+25% TOLERANCE
2653 006636 012637 001274      MOV      (SP)+,TIMEBM ; -25% TOLERANCE
2654
2655      ;*****
2656      ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2657
2658 006642 023727 001256 000764      CMP      TIME,#500.  ;WAS MEASURED TIME AT LEAST 500 MS?
2659 006650 103001                BHS      3S          ;BR IF IT WAS
2660 006652 104055                ERROR    5S          ;REPORT TIMEOUT TOO SHORT
2661
2662      ;*****
2663      ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT
2664
2665 006654 012737 000240 177776 3S:    MOV      #<5*32.>,2#PS ;RESTORE PRIORITY TO 5
2666
2667      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2668
2669 006662 005037 001254      CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
2670 006666 012737 000012 001122      MOV      #RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
2671 006674 060037 001122      ADD      RO,$BDAOR   ;ADD THE I/O BASE ADDRESS
2672 006700 012737 011600 001124      MOV      #MOL!PGM!OPR!DRY,$GDDAT ;COMPARISON CONSTANT
2673 006706 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
2674 006714 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2675 006722 042737 024001 001170      BIC      #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
2676 006730 013737 001170 001164      MOV      $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
2677 006736 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2678 006744 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
2679 006752 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2680 006760 042737 024001 001172      BIC      #PIP!WAL!OM,$TMP3 ;CLEAR DONT CARES
2681 006766 013737 001172 001166      MOV      $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
2682 006774 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2683 007002 023737 001164 001166      CMP      $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2684 007010 001006                BNE      66S        ;BR IF NOT
2685 007012 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?

```

```

2686 007016 001037 BNE 68$ ;BR IF NOT
2687 007020 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2688 007022 000137 007206 JMP 70$ ;BYPASS THE REST OF THE CHECKS
2689 007026 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2690 007034 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2691 007042 113760 001226 000010 MOVVB PORTB,RMCS2(RO) ;SELECT PORT B.
2692 007050 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2693 007054 001414 BEQ 67$ ;BR IF ZERO
2694 007056 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2695 007064 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2696 007072 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A.
2697 007100 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2698 007104 001004 BNE 68$ ;BR IF NOT
2699 007106 012737 177777 001254 67$: MOV #-1,RELEARR ;SET 'RELEASE ERROR' INDICATOR
2700 007114 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
2701 007116 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD51 READ
2702 007124 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
2703 007132 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
2704 007140 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK?
2705 007146 001401 BEQ 69$ ;BR IF OK FROM PORT A.
2706 007150 104007 ERROR 7 ;REPORT ERROR
2707 007152 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMD51 FOR BIT FAILURES - FROM PORT B.
2708 007160 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2709 007166 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
2710 007174 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
2711 007202 001401 BEQ 70$ ;BR IF OK
2712 007204 104007 ERROR 7 ;REPORT ERROR
2713 007206 000240 70$: NOP
2714
2715 ;*****
2716 ;CHECK THE REGISTERS STORED THROUGH PORT A. ALL REGISTERS SHOULD BE ZERO.
2717 ;THE REGISTERS ARE STORED ON THE STACK.
2718
2719 007210 013737 001224 001240 4$: MOV PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2720 007216 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER1
2721 007222 062737 000014 001122 ADD #RMER1,$BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
2722 007230 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER1
2723 007234 001401 BEQ .+4 ;CONTENTS ZERO?
2724 007236 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2725 007240 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMMR1
2726 007244 062737 000024 001122 ADD #RMMR1,$BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
2727 007252 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
2728 007256 001401 BEQ .+4 ;CONTENTS ZERO?
2729 007260 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2730 007262 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
2731 007266 062737 000006 001122 ADD #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
2732 007274 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
2733 007300 001401 BEQ .+4 ;CONTENTS ZERO?
2734 007302 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2735 007304 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
2736 007310 062737 000026 001122 ADD #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
2737 007316 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
2738 007322 001401 BEQ .+4 ;CONTENTS ZERO?
2739 007324 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2740 007326 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMLA
2741 007332 062737 000020 001122 ADD #RMLA,$BDADR ;ADDRESS OF RMLA FOR TYPEOUT

```

```

2742 007340 012637 001126      MOV      (SP)+, $BDDAT      ;CHECK THE STORED CONTENTS OF RMLA
2743 007344 001401              BEQ      .+4                ;CONTENTS ZERO ?
2744 007346 104006              ERROR   6                  ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2745 007350 010037 001122      MOV      R0, $BDAOR        ;BASE ADDRESS FOR REGISTER RMR2
2746 007354 062737 000042 001122      ADD      #RMR2, $BDAOR     ;ADDRESS OF RMR2 FOR TYPEOUT
2747 007362 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMR2
2748 007366 001401              BEQ      .+4                ;CONTENTS ZERO ?
2749 007370 104006              ERROR   6                  ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2750 007372 010037 001122      MOV      R0, $BDADR        ;BASE ADDRESS FOR REGISTER RMOF
2751 007376 062737 000032 001122      ADD      #RMOF, $BDAOR     ;ADDRESS OF RMOF FOR TYPEOUT
2752 007404 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMOF
2753 007410 001401              BEQ      .+4                ;CONTENTS ZERO ?
2754 007412 104006              ERROR   6                  ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2755 007414 010037 001122      MOV      R0, $BDAOR        ;BASE ADDRESS FOR REGISTER RMDC
2756 007420 062737 000034 001122      ADD      #RMDC, $BDAOR     ;ADDRESS OF RMDC FOR TYPEOUT
2757 007426 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMDC
2758 007432 001401              BEQ      .+4                ;CONTENTS ZERO ?
2759 007434 104006              ERROR   6                  ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2760 007436 010037 001122      MOV      R0, $BDAOR        ;BASE ADDRESS FOR REGISTER RMSN
2761 007442 062737 000030 001122      ADD      #RMSN, $BDAOR     ;ADDRESS OF RMSN FOR TYPEOUT
2762 007450 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMSN
2763 007454 001401              BEQ      .+4                ;CONTENTS ZERO ?
2764 007456 104006              ERROR   6                  ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2765 007460 010037 001122      MOV      R0, $BDAOR        ;BASE ADDRESS FOR REGISTER RMEC1
2766 007464 062737 000044 001122      ADD      #RMEC1, $BDAOR    ;ADDRESS OF RMEC1 FOR TYPEOUT
2767 007472 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMEC1
2768 007476 001401              BEQ      .+4                ;CONTENTS ZERO ?
2769 007500 104006              ERROR   6                  ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2770 007502 010037 001122      MOV      R0, $BDAOR        ;BASE ADDRESS FOR REGISTER RMEC2
2771 007506 062737 000046 001122      ADD      #RMEC2, $BDAOR    ;ADDRESS OF RMEC2 FOR TYPEOUT
2772 007514 012637 001126      MOV      (SP)+, $BDDAT     ;CHECK THE STORED CONTENTS OF RMEC2
2773 007520 001401              BEQ      .+4                ;CONTENTS ZERO ?
2774 007522 104006              ERROR   6                  ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2775 007524 000004      SS:      SCOPE              ;LOOP ?

```

```

2776
2777
2778 *****
2779 *TEST 4      PORT 'A' SEIZE/RELEASE TEST
2780 *
2781 *TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
2782 *
2783 * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD51.
2784 *
2785 * B. SET VOLUME VALID AND CLEAR ANY ERROR
2786 *
2787 * C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
2788 * RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
2789 * DRIVE.
2790 *****

```

```

2791 007526
2792 007526 005737 001300      TST     KYBCTL              ;PERFORMING ONLY SINGLE TESTS ?
2793 007532 001406              BEQ     2$                  ;BR IF NOT
2794 007534 100002              BPL     1$                  ;BR IF JUST ENTERED TEST
2795 007536 000137 002676              JMP     EXEC                ;RETURN & GET NEXT TEST NUMBER
2796 007542 012737 177777 001300 1$:      MOV     #-1, KY3CTL        ;SET SINGLE TEST INDICATOR
2797 007550 112737 000004 001102 2$:      MOV     #4, $TSTNM         ;TEST NUMBER

```

```

2798 007556 012737 007600 001106      MOV      #TEST4,$LPADR      ;LOAD LOOP ON TEST ADDRESS
2799 007564 012737 007600 001110      MOV      #TEST4,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
2800 007572 012737 000031 001176      MOV      #25,$TIMES        ;DO 25. ITERATIONS
2801 007600 012706 001100          TEST4:  MOV      #STACK,$SP ;LOAD THE STACK POINTER
2802
2803      ;:*****
2804      ;START THE TIMER
2805
2806 007604 005037 001256          CLR      TIME              ;CLEAR THE ELAPSED TIME COUNTER
2807 007610 012737 003720 001260      MOV      #2000.,WATCH     ;SET WATCH TO 2000 MS
2808
2809      ;:*****
2810      ;SEIZE THE DRIVE AND SET VOLUME VALID
2811
2812      ;SEIZE THE DRIVE THROUGH PORT A
2813
2814 007616 113760 001224 000010      MOVB     PORTA,$RMCS2($RO) ;SELECT PORT A
2815 007624 013737 001224 001242      MOV      PORTA,$SEIZPT    ;STORE SEIZING PORT'S ADDRESS
2816 007632 005060 000012          CLR      $RMS1($RO)       ;WRITE RMS1
2817 007636 013737 001226 001244      MOV      PORTB,$OPPR      ;'OPPOSITE' PORT ADDRESS
2818 007644 012760 000021 000000      MOV      #21,$RMS1($RO)  ;SET VOLUME VALID
2819 007652 005037 001250          CLR      $CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
2820 007656 016037 000012 001126      MOV      $RMS1($RO),$SDDAT ;GET CONTENTS OF RMS1
2821 007664 012737 000012 001122      MOV      $RMS1,$SBOADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2822 007672 060037 001122          ADD     $RO,$SBOADR       ;ADD RHI1 BASE ADDRESS
2823 007676 012737 000100 001124      MOV      #VV,$GDDAT      ;WHAT REGISTER SHOULD BE
2824 007704 013737 001126 001164      MOV      $SDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2825 007712 042737 177677 001164      BIC     #1CVV,$TMP0      ;SAVE SPECIFIED BITS
2826 007720 023737 001124 001164      CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
2827 007726 001414          BEQ     $66$              ;BR IF OK
2828 007730 013737 001126 001174      MOV      $SDDAT,$TMP4     ;COPY 'BAD DATA'
2829 007736 042737 000100 001174      BIC     #VV,$TMP4        ;CLEAR THE MASKED BITS
2830 007744 053737 001174 001124      BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
2831 007752 104013          ERROR  13                ;TYPE MESSAGE 13
2832 007754 005137 001250          COM     $CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2833 007760 000240          66$:  NOP
2834 007762 012760 000040 000010      MOV      #CLR,$RMCS2($RO) ;CLEAR DRIVE
2835
2836      ;:*****
2837
2838      ;RELEASE THE DRIVE FROM PORT A
2839
2840 007770 113760 001224 000010      MOVB     PORTA,$RMCS2($RO) ;SELECT PORT A
2841 007776 013737 001224 001240      MOV      PORTA,$PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2842 010004 012760 000013 000000      MOV      #13,$RMS1($RO)  ;ISSUE RELEASE THROUGH PORT A
2843
2844      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2845
2846 010012 005037 001254          CLR     $RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
2847 010016 012737 000012 001122      MOV      #RMS1,$SBOADR    ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
2848 010024 060037 001122          ADD     $RO,$SBOADR       ;ADD THE I/O BASE ADDRESS
2849 010030 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2850 010036 113760 001224 000010      MOVB     PORTA,$RMCS2($RO) ;SELECT PORT A.
2851 010044 016037 000012 001170      MOV      $RMS1($RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2852 010052 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
2853 010060 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'

```


E05

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 56
T4 PORT 'A' SEIZE/RELEASE TEST

SEQ 0056

2854	010066	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
2855	010074	113760	001226	000010		MOV	PORTB, RMCS2(RO)	;SELECT PORT B.
2856	010102	016037	000012	001172		MOV	RMDS1(RO), \$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
2857	010110	042737	024001	001172		BIC	#PIP!WAL!OM,\$TMP3	;CLEAR DONT CARES
2858	010116	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
2859	010124	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
2860	010132	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2861	010140	001006				BNE	68\$;BR IF NOT
2862	010142	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2863	010146	001037				BNE	70\$;BR IF NOT
2864	010150	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2865	010152	000137	010336			JMP	72\$;BYPASS THE REST OF THE CHECKS
2866	010156	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2867	010164	013737	001226	001240		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2868	010172	113760	001226	000010		MOV	PORTB, RMCS2(RO)	;SELECT PORT B.
2869	010200	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
2870	010204	001414				BEQ	69\$;BR IF ZERO
2871	010206	013737	001224	001240		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2872	010214	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
2873	010222	113760	001224	000010		MOV	PORTA, RMCS2(RO)	;SELECT PORT A.
2874	010230	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
2875	010234	001004				BNE	70\$;BR IF NOT
2876	010236	012737	177777	001254	69\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
2877	010244	104022				ERROR	22	;TYPE ERROR MESSAGE 22
2878	010246	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMDS1 READ
2879	010254	013737	001224	001240		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
2880	010262	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
2881	010270	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
2882	010276	001401				BEQ	71\$;BR IF OK FROM PORT A.
2883	010300	104007				ERROR	7	;REPORT ERROR
2884	010302	013737	001172	001126	71\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
2885	010310	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
2886	010316	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
2887	010324	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
2888	010332	001401				BEQ	72\$;BR IF OK
2889	010334	104007				ERROR	7	;REPORT ERROR
2890	010336	000240			72\$:	NOP		
2891	010340	005737	001254			TST	RELERR	;DID DRIVE RETURN TO NEUTRAL ?
2892	010344	001402				BEQ	:+6	;BR IF IN NEUTRAL
2893	010346	000137	010622			JMP	1\$;GO WAIT FOR DRIVE TO TIMEOUT
2894	010352	113760	001224	000010		MOV	PORTA, RMCS2(RO)	;SELECT PORT A
2895	010360	013737	001224	001240		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2896	010366	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
2897	010372	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
2898	010400	012737	000012	001122		MOV	#RMDS1,\$BDDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
2899	010406	060037	001122			ADD	RO,\$BDDADR	;ADD RH11 BASE ADDRESS
2900	010412	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
2901	010416	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
2902	010424	042737	077777	001164		BIC	#!CATA,\$TMP0	;SAVE SPECIFIED BITS
2903	010432	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
2904	010440	001414				BEQ	73\$;BR IF OK
2905	010442	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
2906	010450	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
2907	010456	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
2908	010464	104017				ERROR	17	;TYPE MESSAGE 17
2909	010466	005137	001250			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR

F05

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 57
T4 PORT 'A' SEIZE/RELEASE TEST

SEQ 0057

2910	010472	000240		
2911	010474	113760	001226	000010
2912	010502	013737	001226	001240
2913	010510	005037	001250	
2914	010514	016037	000012	001126
2915	010522	012737	000012	001122
2916	010530	060037	001122	
2917	010534	005037	001124	
2918	010540	013737	001126	001164
2919	010546	042737	077777	001164
2920	010554	023737	001124	001164
2921	010562	001414		
2922	010564	013737	001126	001174
2923	010572	042737	100000	001174
2924	010600	053737	001174	001124
2925	010606	104017		
2926	010610	005137	001250	
2927	010614	000240		
2928	010616	000137	010654	
2929				
2930				
2931				
2932				
2933				
2934	010622			
2935	010622	113760	001226	000010
2936	010630	013737	001226	001240
2937	010636	005760	000012	
2938	010642	001004		
2939	010644	005737	001260	
2940	010650	001364		
2941	010652	104036		
2942	010654	000004		
2943				
2944				
2945				
2946				
2947				
2948				
2949				
2950				
2951				
2952				
2953				
2954				
2955				
2956				
2957				
2958	010656			
2959	010656	005737	001300	
2960	010662	001406		
2961	010664	100002		
2962	010666	000137	002676	
2963	010672	012737	177777	001300
2964	010700	112737	000005	001102
2965	010706	012737	010730	001106

```

73$:  NOP
      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
      MOV  PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
      MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
      MOV  #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
      ADD  RO, $BDAOR ; ADD RHI1 BASE ADDRESS
      CLR  $GDDAT ; WHAT REGISTER SHOULD BE
      MOV  $BDDAT, $TMPO ; MOVE REGISTER CONTENTS TO '$TMPO'
      BIC  #1CATA, $TMPO ; SAVE SPECIFIED BITS
      CMP  $GDDAT, $TMPO ; COMPARE THE BITS
      BEQ  75$ ; BR IF OK
      MOV  $BDDAT, $TMP4 ; COPY 'BAD DATA'
      BIC  #ATA, $TMP4 ; CLEAR THE MASKED BITS
      BIS  $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
      ERROR 17 ; TYPE MESSAGE 17
      COM  CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
75$:  NOP
      JMP  2$ ; GO CHECK FOR LOOP ON ERROR

;*****
;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
;TO RELEASE THE DRIVE
1$:  MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
      MOV  PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      TST  RMDS1(RO) ; WAIT FOR TIMEOUT TO RELEASE DRIVE
      BNE  2$ ; BR WHEN DRIVE RELEASED
      TST  WATCH ; CHECK THE WATCH
      BNE  1$ ; BR IF NOT ZERO
      ERROR 36 ; NO TIMEOUT WITHIN 2 SECONDS
2$:  SCOPE ; LOOP ?

;*****
*TEST 5 PORT 'B' SEIZE/RELEASE TEST
*
*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
*
* B. SET VOLUME VALID AND CLEAR ANY ERROR
*
* C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
* DRIVE.
;*****
↑ST5: 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 #5, $STNM ; TEST NUMBER
      MOV  #TEST5, $LPAOR ; LOAD LOOP ON TEST ADDRESS

```

G05

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 58
T5 PORT 'B' SEIZE/RELEASE TEST

SEQ 0058

```

2966 010714 012737 010730 001110      MOV      #TESTS,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
2967 010722 012737 000031 001176      MOV      #25,$TIMES        ;DO 25. ITERATIONS
2968 010730 012706 001100      TESTS:  MOV      #STACK,SP  ;LOAD THE STACK POINTER
2969
2970      ;:*****
2971      ;START THE TIMER
2972
2973 010734 005037 001256      CLR      TIME              ;CLEAR THE ELAPSED TIME COUNTER
2974 010740 012737 003720 001260      MOV      #2000.,WATCH     ;SET WATCH TO 2000 MS
2975
2976      ;:*****
2977      ;SEIZE THE DRIVE AND SET VOLUME VALID
2978
2979      ;SEIZE THE DRIVE THROUGH PORT B
2980
2981 010746 113760 001226 000010      MOV      PORTB, RMCS2(RO)  ;SELECT PORT B
2982 010754 013737 001226 001242      MOV      PORTB, SEIZPT    ;STORE SEIZING PORT'S ADDRESS
2983 010762 005060 000012      CLR      RMDS1(RO)        ;WRITE RMDS1
2984 010766 013737 001224 001244      MOV      PORTA, OPPRT     ;'OPPOSITE' PORT ADDRESS
2985 010774 012760 000021 000000      MOV      #21, RMCS1(RO)   ;SET VOLUME VALID
2986 011002 005037 001250      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2987 011006 016037 000012 001126      MOV      RMDS1(RO), $BODAT ;GET CONTENTS OF RMDS1
2988 011014 012737 000012 001122      MOV      #RMDS1,$BODADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2989 011022 060037 001122      ADD      RO,$BODADR       ;ADD RHI1 BASE ADDRESS
2990 011026 012737 000100 001124      MOV      #VV,$GDDAT      ;WHAT REGISTER SHOULD BE
2991 011034 013737 001126 001164      MOV      $BODAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2992 011042 042737 177677 001164      BIC      #CVV,$TMP0      ;SAVE SPECIFIED BITS
2993 011050 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
2994 011056 001414      BEQ      66$             ;BR IF OK
2995 011060 013737 001126 001174      MOV      $BODAT,$TMP4    ;COPY 'BAD DATA'
2996 011066 042737 000100 001174      BIC      #VV,$TMP4       ;CLEAR THE MASKED BITS
2997 011074 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2998 011102 104013      ERROR   13              ;TYPE MESSAGE 13
2999 011104 005137 001250      COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3000 011110 000240      66$:   NOP
3001 011112 012760 000040 000010      MOV      #CLR,RMCS2(RO)   ;CLEAR DRIVE
3002
3003      ;:*****
3004
3005      ;RELEASE THE DRIVE FROM PORT B
3006
3007 011120 113760 001226 000010      MOV      PORTB, RMCS2(RO)  ;SELECT PORT B
3008 011126 013737 001226 001240      MOV      PORTB, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3009 011134 012760 000013 000000      MOV      #13, RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT B
3010
3011      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3012
3013 011142 005037 001254      CLR      RELERR           ;CLEAR THE 'RELEASE ERROR' INDICATOR
3014 011146 012737 000012 001122      MOV      #RMDS1,$BODADR   ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3015 011154 060037 001122      ADD      RO,$BODADR       ;ADD THE I/O BASE ADDRESS
3016 011160 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
3017 011166 113760 001224 000010      MOV      PORTA, RMCS2(RO)  ;SELECT PORT A.
3018 011174 016037 000012 001170      MOV      RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3019 011202 042737 024001 001170      BIC      #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
3020 011210 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3021 011216 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```

H05

CZRMGBD RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 59
TS PORT 'B' SEIZE/RELEASE TEST

SEG 0059

3022	011224	113760	001226	000010		MOV B	PORTB, RMCS2(R0)	; SELECT PORT B.
3023	011232	016037	000012	001172		MOV	RMDS1(R0), STMP3	; GET THE DRIVE STATUS REGISTER FROM PORT B.
3024	011240	042737	024001	001172		BIC	#PIP!WAL!OM, STMP3	; CLEAR DONT CARES
3025	011246	013737	001172	001166		MOV	STMP3, STMP1	; COPY IT INTO 'STMP1'
3026	011254	042737	100100	001166		BIC	#ATA!VV, STMP1	; CLEAR PORT DEPENDENT BITS FROM THE COPY
3027	011262	023737	001164	001166		CMP	STMP0, STMP1	; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3028	011270	001006				BNE	68\$; BR IF NOT
3029	011272	005737	001164			TST	STMP0	; REGISTERS ARE THE SAME: ARE THEY ZERO ?
3030	011276	001037				BNE	70\$; BR IF NOT
3031	011300	104046				ERROR	46	; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3032	011302	000137	011466			JMP	72\$; BYPASS THE REST OF THE CHECKS
3033	011306	013737	001170	001126	68\$:	MOV	STMP2, \$BDDAT	; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3034	011314	013737	001226	001240		MOV	PORTB, PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3035	011322	113760	001226	000010		MOV B	PORTB, RMCS2(R0)	; SELECT PORT B.
3036	011330	005737	001164			TST	STMP0	; SEE IF STATUS EQ 0 FROM PORT A.
3037	011334	001414				BEQ	69\$; BR IF ZERO
3038	011336	013737	001224	001240		MOV	PORTA, PTNBR	; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3039	011344	013737	001172	001126		MOV	STMP3, \$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3040	011352	113760	001224	000010		MOV B	PORTA, RMCS2(R0)	; SELECT PORT A.
3041	011360	005737	001166			TST	STMP1	; SEE IF STATUS EQ ZERO FROM PORT B.
3042	011364	001004				BNE	70\$; BR IF NOT
3043	011366	012737	177777	001254	69\$:	MOV	#-1, RELERR	; SET 'RELEASE ERROR' INDICATOR
3044	011374	104022				ERROR	22	; TYPE ERROR MESSAGE 22
3045	011376	013737	001170	001126	70\$:	MOV	STMP2, \$BDDAT	; LOOK FOR BIT FAILURES WHEN RMDS1 READ
3046	011404	013737	001224	001240		MOV	PORTA, PTNBR	; CHANGE PORT NUMBER
3047	011412	042737	100100	001126		BIC	#ATA!VV, \$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
3048	011420	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; ALL BITS OK ?
3049	011426	001401				BEQ	71\$; BR IF OK FROM PORT A.
3050	011430	104007				ERROR	7	; REPORT ERROR
3051	011432	013737	001172	001126	71\$:	MOV	STMP3, \$BDDAT	; CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3052	011440	013737	001226	001240		MOV	PORTB, PTNBR	; CHANGE PORT NUMBER
3053	011446	042737	100100	001126		BIC	#ATA!VV, \$BDDAT	; DON'T CHECK ATTN BIT OR VV BIT
3054	011454	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	; SEE IF READ OK FROM PORT B.
3055	011462	001401				BEQ	72\$; BR IF OK
3056	011464	104007				ERROR	7	; REPORT ERROR
3057	011466	000240			72\$:	NOP		
3058	011470	005737	001254			TST	RELERR	; DID DRIVE RETURN TO NEUTRAL ?
3059	011474	001402				BEQ	+6	; BR IF IN NEUTRAL
3060	011476	000137	011752			JMP	1\$; GO WAIT FOR DRIVE TO TIMEOUT
3061	011502	113760	001226	000010		MOV B	PORTB, RMCS2(R0)	; SELECT PORT B
3062	011510	013737	001226	001240		MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3063	011516	005037	001250			CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
3064	011522	016037	000012	001126		MOV	RMDS1(R0), \$BDDAT	; GET CONTENTS OF RMDS1
3065	011530	012737	000012	001122		MOV	#RMDS1, \$BDDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
3066	011536	060037	001122			ADD	R0, \$BDDADR	; ADD RHI1 BASE ADDRESS
3067	011542	005037	001124			CLR	\$GDDAT	; WHAT REGISTER SHOULD BE
3068	011546	013737	001126	001164		MOV	\$BDDAT, STMP0	; MOVE REGISTER CONTENTS TO 'STMP0'
3069	011554	042737	077777	001164		BIC	#!CATA, STMP0	; SAVE SPECIFIED BITS
3070	011562	023737	001124	001164		CMP	\$GDDAT, STMP0	; COMPARE THE BITS
3071	011570	001414				BEQ	73\$; BR IF OK
3072	011572	013737	001126	001174		MOV	\$BDDAT, STMP4	; COPY 'BAD DATA'
3073	011600	042737	100000	001174		BIC	#ATA, STMP4	; CLEAR THE MASKED BITS
3074	011606	053737	001174	001124		BIS	STMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3075	011614	104017				ERROR	17	; TYPE MESSAGE 17
3076	011616	005137	001250			COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
3077	011622	000240			73\$:	NOP		

```

3078 011624 113760 001224 000010      MOV  PORTA, RMCS2(RO) ;SELECT PORT A
3079 011632 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3080 011640 005037 001250      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3081 011644 016037 000012 001126      MOV  RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
3082 011652 012737 000012 001122      MOV  $RMDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3083 011660 060037 001122      ADD  RO, $BDAOR ;ADD RHI1 BASE ADDRESS
3084 011664 005037 001124      CLR  $GDDAT ;WHAT REGISTER SHOULD BE
3085 011670 013737 001126 001164      MOV  $BDDAT, $TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
3086 011676 042737 077777 001164      BIC  $ICATA, $TMPO ;SAVE SPECIFIED BITS
3087 011704 023737 001124 001164      CMP  $GDDAT, $TMPO ;COMPARE THE BITS
3088 011712 001414      BEQ  75$ ;BR IF OK
3089 011714 013737 001126 001174      MOV  $BDDAT, $TMP4 ;COPY 'BAD DATA'
3090 011722 042737 100000 001174      BIC  $ATA, $TMP4 ;CLEAR THE MASKED BITS
3091 011730 053737 001174 001124      BIS  $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3092 011736 104017      ERROR 17 ;TYPE MESSAGE 17
3093 011740 005137 001250      COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3094 011744 000240      NOP
3095 011746 000137 012004      JMP  25 ;GO CHECK FOR LOOP ON ERROR
3096
3097
3098
3099
3100

```

```

*****
: IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEJUT
: TO RELEASE THE DRIVE

```

```

3101 011752      1S:
3102 011752 113760 001224 000010      MOV  PORTA, RMCS2(RO) ;SELECT PORT A
3103 011760 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3104 011766 005760 000012      TST  RMDS1(RO) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3105 011772 001004      BNE  25 ;BR WHEN DRIVE RELEASED
3106 011774 005737 001260      TST  WATCH ;CHECK THE WATCH
3107 012000 001364      BNE  15 ;BR IF NOT ZERO
3108 012002 104036      ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
3109 012004 000004      SCOPE ;LOOP ?
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120

```

```

*****
: *TEST 6 PORT 'A' NEUTRAL/RELEASE TEST
: *
: *TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
: *
: * A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
: * NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
: *

```

```

3121 012006      1S:
3122 012006 005737 001300      TST  KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
3123 012012 001406      BEQ  25 ;BR IF NOT
3124 012014 100002      BPL  15 ;BR IF JUST ENTERED TEST
3125 012016 000137 002676      JMP  EXEC ;RETURN & GET NEXT TEST NUMBER
3126 012022 012737 177777 001300      MOV  #-1, KYBCTL ;SET SINGLE TEST INDICATOR
3127 012030 112737 000006 001102      2S: MOV  $6, $TSTNM ;TEST NUMBER
3128 012036 012737 012060 001106      MOV  $TEST6, $LPADR ;LOAD LOOP ON TEST ADDRESS
3129 012044 012737 012060 001110      MOV  $TEST6, $LPERR ;LOAD LOOP ON ERROR ADDRESS
3130 012052 012737 000031 001176      MOV  $25, $TIMES ;DO 25 ITERATIONS
3131 012060 012706 001100      TEST6: MOV  $STACK, SP ;LOAD THE STACK POINTER
3132 012064 113760 001224 000010      MOV  PORTA, RMCS2(RO) ;SELECT PORT A
3133 012072 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

3134 012100 013737 001224 001242      MOV      PORTA,SEIZPT      ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3135
3136      ;*****
3137      ;ISSUE A RELEASE COMMAND
3138
3139 012106 012760 000013 000000      MOV      #13,RMCS1(RO)    ;ISSUE A RELEASE COMMAND
3140
3141      ;*****
3142      ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
3143
3144
3145      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3146
3147 012114 005037 001254      CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3148 012120 012737 000012 001122      MOV      #RMDS1,$BDDADR   ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3149 012126 060037 001122      ADD      RO,$BDDADR       ;ADD THE I/O BASE ADDRESS
3150 012132 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3151 012140 113760 001224 000010      MCVB     PORTA,RMCS2(RO)  ;SELECT PORT A.
3152 012146 016037 000012 001170      MOV      RMDS1(RO),STMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3153 012154 042737 024001 001170      BIC      #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
3154 012162 013737 001170 001164      MOV      STMP2,STMP0      ;COPY IT INTO 'STMP0'
3155 012170 042737 100100 001164      BIC      #ATA!VV,STMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3156 012176 113760 001226 000010      MOVVB    PORTB,RMCS2(RO)  ;SELECT PORT B.
3157 012204 016037 000012 001172      MOV      RMDS1(RO),STMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3158 012212 042737 024001 001172      BIC      #PIP!WRL!OM,STMP3 ;CLEAR DONT CFRES
3159 012220 013737 001172 001166      MOV      STMP3,STMP1     ;COPY IT INTO 'STMP1'
3160 012226 042737 100100 001166      BIC      #ATA!VV,STMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3161 012234 023737 001164 001166      CMP      STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3162 012242 001006      BNE      64$             ;BR IF NOT
3163 012244 005737 001164      TST      STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3164 012250 001045      BNE      66$             ;BR IF NOT
3165 012252 104046      ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3166 012254 000137 012454      JMP      68$             ;BYPASS THE REST OF THE CHECKS
3167 012260 013737 001170 001126 64$:    MOV      STMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3168 012266 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3169 012274 113760 001226 000010      MOVVB    PORTB,RMCS2(RO) ;SELECT PORT B.
3170 012302 005737 001164      TST      STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3171 012306 001414      BEQ      65$             ;BR IF ZERO
3172 012310 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3173 012316 013737 001172 001126      MOV      STMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3174 012324 113760 001224 000010      MOVVB    PORTA,RMCS2(RO) ;SELECT PORT A.
3175 012332 005737 001166      TST      STMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3176 012336 001012      BNE      66$             ;BR IF NOT
3177 012340 012737 177777 001254 65$:    MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3178 012346 012760 000011 000000      MOV      #11,RMCS1(RO)   ;CLEAR THE DRIVE
3179 012354 012760 000013 000000      MOV      #13,RMCS1(RO)   ;RELEASE THE DRIVE
3180 012362 104030      ERROR    30             ;TYPE ERROR MESSAGE 30
3181 012364 013737 001170 001126 66$:    MOV      STMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3182 012372 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
3183 012400 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
3184 012406 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
3185 012414 001401      BEQ      67$             ;BR IF OK FROM PORT A.
3186 012416 104007      ERROR    7              ;REPORT ERROR
3187 012420 013737 001172 001126 67$:    MOV      STMP3,$BDDAT    ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3188 012426 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
3189 012434 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT

```

K05

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 62
T6 PORT 'A' NEUTRAL/RELEASE TEST

SEG 0062

```

3190 012442 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
3191 012450 001401                    BEQ      685           ;BR IF OK
3192 012452 104007                    ERROR    7            ;REPORT ERROR
3193 012454 000240                    NOP
3194 012456 000004                    SCOPE          ;LOOP ?
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205 012460
3206 012460 005737 001300      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
3207 012464 001406                    BEQ      25           ;BR IF NOT
3208 012466 100002                    BPL      1$          ;BR IF JUST ENTERED TEST
3209 012470 000137 002676      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
3210 012474 012737 177777 001300 1$: MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3211 012502 112737 000007 001102 2$: MOVB   #7,$STNM   ;TEST NUMBER
3212 012510 012737 012532 001106      MOV      #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
3213 012516 012737 012532 001110      MOV      #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3214 012524 012737 000031 001176      MOV      #25,$TIMES  ;DO 25 ITERATIONS
3215 012532 012706 001100      TEST7: MOV     $STACK,SP ;LOAD THE STACK POINTER
3216 012536 113760 001226 000010      MOVB    PORTB,$RMCS2(RO) ;SELECT PORT B
3217 012544 013737 001226 001240      MOV     PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3218 012552 013737 001226 001242      MOV     PORTB,$SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3219
3220
3221
3222
3223 012560 012760 000013 000000      MOV     #13,$RMCS1(RO) ;ISSUE A RELEASE COMMAND
3224
3225
3226
3227
3228
3229
3230
3231 012566 005037 001254                    CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
3232 012572 012737 000012 001122      MOV     $RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3233 012600 060037 001122      ADD     RO,$BDAOR    ;ADD THE I/O BASE ADDRESS
3234 012604 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3235 012612 113760 001224 000010      MOVB   PORTA,$RMCS2(RO) ;SELECT PORT A.
3236 012620 016037 000012 001170      MOV     $RMDS1(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3237 012626 042737 024001 001170      BIC    #PIP!WAL!OM,$STMP2 ;CLEAR DONT CARES
3238 012634 013737 001170 001164      MOV     $STMP2,$STMPD ;COPY IT INTO 'STMPD'
3239 012642 042737 100100 001164      BIC    #ATA!VV,$STMPD ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3240 012650 113760 001226 000010      MOVB   PORTB,$RMCS2(RO) ;SELECT PORT B.
3241 012656 016037 000012 001172      MOV     $RMDS1(RO),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3242 012664 042737 024001 001172      BIC    #PIP!WAL!OM,$STMP3 ;CLEAR DONT CARES
3243 012672 013737 001172 001166      MOV     $STMP3,$STMP1 ;COPY IT INTO 'STMP1'
3244 012700 042737 100100 001166      BIC    #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3245 012706 023737 001164 001166      CMP     $STMPD,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```

```

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

```

```

↑ST7:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      25           ;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  #7,$STNM   ;TEST NUMBER
MOV      #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV      #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV      #25,$TIMES  ;DO 25 ITERATIONS
TEST7: MOV  $STACK,SP ;LOAD THE STACK POINTER
MOVB    PORTB,$RMCS2(RO) ;SELECT PORT B
MOV     PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV     PORTB,$SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

```

```

*****
;ISSUE A RELEASE COMMAND

```

```

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

```

```

*****
;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

```

CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV     $RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
ADD     RO,$BDAOR    ;ADD THE I/O BASE ADDRESS
MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
MOVB   PORTA,$RMCS2(RO) ;SELECT PORT A.
MOV     $RMDS1(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
BIC    #PIP!WAL!OM,$STMP2 ;CLEAR DONT CARES
MOV     $STMP2,$STMPD ;COPY IT INTO 'STMPD'
BIC    #ATA!VV,$STMPD ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB   PORTB,$RMCS2(RO) ;SELECT PORT B.
MOV     $RMDS1(RO),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
BIC    #PIP!WAL!OM,$STMP3 ;CLEAR DONT CARES
MOV     $STMP3,$STMP1 ;COPY IT INTO 'STMP1'
BIC    #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP     $STMPD,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```


LOS

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 63
T? PORT 'B' NEUTRAL/RELEASE TEST

SEQ 0063

3246	012714	001006				BNE	64\$;BR IF NOT
3247	012716	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3248	012722	001045				BNE	66\$;BR IF NOT
3249	012724	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3250	012726	000137	013126			JMP	68\$;BYPASS THE REST OF THE CHECKS
3251	012732	013737	001170	001126	64\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3252	012740	013737	001226	001240		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3253	012746	113760	001226	000010		MOV	PORTB,RMCS2(RO)		;SELECT PORT B.
3254	012754	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
3255	012760	001414				BEQ	65\$;BR IF ZERO
3256	012762	013737	001224	001240		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3257	012770	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
3258	012776	113760	001224	000010		MOV	PORTA,RMCS2(RO)		;SELECT PORT A.
3259	013004	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
3260	013010	001012				BNE	66\$;BR IF NOT
3261	013012	012737	177777	001254	65\$:	MOV	#-1,RELEA		;SET 'RELEASE ERROR' INDICATOR
3262	013020	012760	000011	000000		MOV	#11,RMCS1(RO)		;CLEAR THE DRIVE
3263	013026	012760	000013	000000		MOV	#13,RMCS1(RO)		;RELEASE THE DRIVE
3264	013034	104030				ERROR	30		;TYPE ERROR MESSAGE 3U
3265	013036	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3266	013044	013737	001224	001240		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER
3267	013052	042737	100000	001126		BIC	#ATA,\$BDDAT		;DON'T CHECK THE ATTN BIT
3268	013060	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;ALL BITS OK ?
3269	013066	001401				BEQ	67\$;BR IF OK FROM PORT A.
3270	013070	104007				ERROR	7		;REPORT ERROR
3271	013072	013737	001172	001126	67\$:	MOV	\$TMP3,\$BDDAT		;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3272	013100	013737	001226	001240		MOV	PORTB,PTNBR		;CHANGE PORT NUMBER
3273	013106	042737	100000	001126		BIC	#ATA,\$BDDAT		;DON'T CHECK THE ATTN BIT
3274	013114	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;SEE IF READ OK FROM PORT B.
3275	013122	001401				BEQ	68\$;BR IF OK
3276	013124	104007				ERROR	7		;REPORT ERROR
3277	013126	000240			68\$:	NOP			
3278	013130	000004				SCOPE			;LOOP ?

3279									
3280									
3281									
3282									
3283									
3284									
3285									
3286									
3287									
3288									
3289									
3290									
3291									
3292									
3293									
3294									
3295									
3296									
3297									
3298									
3299									
3300	013132								
3301	013132	005737	001300			TST	KYBCTL		;PERFORMING ONLY SINGLE TESTS ?

```

*****
*TEST 10 PORT 'A' RELEASE INTERFERENCE TEST
*
*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
* IS SEIZED BY THE OTHER PORT.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
*
* 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.
*****

```


MOS

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 64
T10 PORT 'A' RELEASE INTERFERENCE TEST

SEG 0064

```

3302 013136 001406          BEQ      25          ;BR IF NOT
3303 013140 100002          BPL      15          ;BR IF JUST ENTERED TEST
3304 013142 000137 002676    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
3305 013144 012737 177777    001300 15:      MOV      #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
3306 013154 112737 000010 001102 25:      MOVVB   #10,$STSTM   ;TEST NUMBER
3307 013162 012737 013204 001106      MOV      #TEST10,$LPADR ;LOAD LOOP ON TEST ADDRESS
3308 013170 012737 013204 001110      MOV      #TEST10,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3309 013176 012737 000031 001176      MOV      #25,$TIMES    ;DO 25 ITERATIONS
3310 013204 012706 001100      TEST10: MOV     #STACK,SP ;LOAD THE STACK POINTER
3311
3312          ;CLEAR ATTENTION BITS FOR BOTH PORTS
3313
3314 013210 113760 001224 000010      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT #A
3315 013216 005060 000012          CLR      RMD51(RO)      ;SEIZE THE DRIVE
3316 013222 012760 000011 000000      MOV      #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
3317 013230 012760 000013 000000      MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
3318 013236 113760 001226 000010      MOVVB   PORTB, RMCS2(RO) ;SELECT PORT #B
3319 013244 005060 000012          CLR      RMD51(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
3320 013250 012760 000011 000000      MOV      #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
3321 013256 012760 000013 000000      MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
3322
3323          ;*****
3324
3325          ;SEIZE THE DRIVE THROUGH PORT B
3326
3327 013264 113760 001226 000010      MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
3328 013272 013737 001226 001242      MOV      PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
3329 013300 005060 000012          CLR      RMD51(RO)      ;WRITE RMD51
3330 013304 113760 001224 000010      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
3331 013312 013737 001224 001240      MOV      PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3332 013320 013737 001224 001244      MOV      PORTA, OPPRT   ;'OPPOSITE' PORT ADDRESS
3333 013326 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
3334 013334 010037 001122          MOV      RO, $BDAOR     ;R#11 BASE ADDRESS
3335 013340 062737 000012 001122      ADD      #RMD51, $BDAOR ;GENERATE BAD REGISTER ADDRESS
3336 013346 045037 001124          CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
3337 013352 023737 001124 001126      CMP      $GDDAT, $BDDAT ;IS THE REGISTER ZERO
3338 013360 001403          BEQ      645          ;BR IF IT IS
3339 013362 104004          ERROR   4            ;REPORT THE ERROR
3340 013364 000137 014374          JMP      15           ;BYPASS REST OF THE SUBTEST
3341 013370
3342 013370 113760 001226 000010 645:      MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
3343 013376 013737 001226 001240      MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3344 013404 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
3345 013412 042737 020001 001126      BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
3346 013420 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
3347 013426 013737 001124 001166      MOV      $GDDAT, $TMP1  ;USE GOOD DATA AS A MASK
3348 013434 005137 001166          COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
3349 013440 013737 001126 001164      MOV      $BDDAT, $TMP0  ;SAVE THE ACTUAL STATUS
3350 013446 043737 001166 001164      BIC      $TMP1, $TMP0   ;CLEAR UNWANTED BITS
3351 013454 023737 001124 001164      CMP      $GDDAT, $TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
3352 013462 001401          BEQ      655          ;BR IF THEY ARE
3353 013464 104005          ERROR   5            ;REPORT THE ERROR
3354 013466 000240 655:      NOP
3355
3356          ;*****
3357          ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

```

N05

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 65
T10 PORT 'A' RELEASE INTERFERENCE TEST

SEQ 0065

```

3358
3359 013470 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
3360 013476 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3361 013504 012760 000013 000000      MOV   #13, RMCS1(RO) ; ISSUE A RELEASE COMMAND THROUGH PORT A
3362
3363 ;*****
3364 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
3365
3366 013512 005037 001250      CLR   CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
3367 013516 016037 000012 001126      MOV   RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
3368 013524 012737 000012 001122      MOV   #RMDS1, $B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
3369 013532 060037 001122      ADD   RO, $B0ADR ; ADD RHI1 BASE ADDRESS
3370 013536 005037 001124      CLR   $GDDAT ; WHAT REGISTER SHOULD BE
3371 013542 023737 001124 001126      CMP   $GDDAT, $BDDAT ; IS THE REGISTER OK ?
3372 013550 001403      BEQ   66$ ; BR IF OK
3373 013552 104010      ERROR 10 ; REPORT THE ERROR
3374 013554 005137 001250      COM   CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
3375 013560 016037 000000 001126 66$: MOV   RMCS1(RO), $BDDAT ; GET THE CONTENTS OF RMCS1
3376 013566 012737 000000 001122      MOV   #RMCS1, $B0ADR ; FORM ADDRESS OF REGISTER
3377 013574 060037 001122      ADD   RO, $B0ADR ; ADDRESS BASE
3378 013600 032737 020000 001126      BIT   #MCPE, $BDDAT ; IS 'MCPE' SET ?
3379 013606 001404      BEQ   67$ ; BR IF NOT
3380 013610 104011      ERROR 11 ; REPORT THE ERROR
3381 013612 012760 040000 000000      MOV   #TRE, RMCS1(RO) ; CLEAR 'MCPE'
3382 013620 000240      NOP
3383 013622 005737 001250      TST   CKERR ; WAS RMDS1 NON ZERO ?
3384 013626 001402      BEQ   +6 ; CONTENTS OF RMDS1 SEEN BY PORT A
3385 013630 000137 014374      JMP   IS ; DRIVE IN NEUTRAL, BYPASS REST OF TEST
3386
3387 ;*****
3388
3389 ;RELEASE THE DRIVE FROM PORT B
3390
3391 013634 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
3392 013642 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3393 013650 012760 000013 000000      MOV   #13, RMCS1(RO) ; ISSUE RELEASE THROUGH PORT B
3394
3395 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
3396
3397 013656 005037 001254      CLR   RELERR ; CLEAR 'RELEASE ERROR' INDICATOR
3398 013662 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ; COMPARISON CONSTANT
3399 013670 012737 000012 001122      MOV   #RMDS1, $B0ADR ; REGISTER ADDRESS INCREMENT
3400 013676 060037 001122      ADD   RO, $B0ADR ; REGISTER BASE ADDRESS FOR TYPEOUT
3401 013702 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
3402 013710 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3403 013716 016037 000012 001164      MOV   RMDS1(RO), $TMPD ; READ STATUS REGISTER FROM PORT A
3404 013724 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
3405 013732 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3406 013740 016037 000012 001126      MOV   RMDS1(RO), $BDDAT ; DRIVE STATUS FROM PORT B
3407 013746 001404      BEQ   68$ ; BR IF STATUS FROM PORT B ZERO
3408 013750 005737 001164      TST   $TMPD ; IS STATUS FROM PORT A ZERO ?
3409 013754 001401      BEQ   68$ ; BR IF ZERO
3410 013756 104031      ERROR 31 ; REPORT DRIVE IN NEUTRAL
3411 013760 013737 001164 001126 68$: MOV   $TMPD, $BDDAT ; CHECK STATUS FROM PORT A
3412 013766 013737 001224 001240      MOV   PORTA, PTNBR ; CHANGE PORT ADDRESS FOR TYPEOUT
3413 013774 023737 001124 001126      CMP   $GDDAT, $BDDAT ; COMPARE WITH CONSTANT

```

```

3414 014002 001401      BEQ      69$      ;BR IF OK
3415 014004 104027      ERROR    27      ;REPORT REGISTER ERROR
3416 014006 000240      69$:      NOP
3417
3418 ;RELEASE THE DRIVE FROM PORT A
3419
3420 014010 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
3421 014016 013737 001224 001240      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3422 014024 012760 000013 000000      MOV      #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
3423
3424 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3425
3426 014032 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3427 014036 012737 000012 001122      MOV      #RMOS1, $BDDADR ;FORM THE ADDRESS OF RMOS1 FOR TYPEOUT
3428 014044 060037 0C1122      ADD      RO, $BDDADR ;ADD THE I/O BASE ADDRESS
3429 014050 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3430 014056 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A.
3431 014064 016037 000012 001170      MOV      RMOS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3432 014072 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
3433 014100 013737 001170 001164      MOV      $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
3434 014106 042737 100100 001164      BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3435 014114 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
3436 014122 016037 000012 001172      MOV      RMOS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3437 014130 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
3438 014136 013737 001172 001166      MOV      $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
3439 014144 042737 100100 001166      BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3440 014152 023737 001164 001166      CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3441 014160 001006      BNE      70$      ;BR IF NOT
3442 014162 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3443 014166 001045      BNE      72$      ;BR IF NOT
3444 014170 104046      ERROR    46      ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3445 014172 000137 014372      JMP      74$      ;BYPASS THE REST OF THE CHECKS
3446 014176 013737 001170 001126 70$:      MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3447 014204 013737 001226 001240      MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3448 014212 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
3449 014220 005737 001164      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3450 014224 001414      BEQ      71$      ;BR IF ZERO
3451 014226 013737 001224 001240      MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3452 014234 013737 001172 001126      MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3453 014242 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A.
3454 014250 005737 001166      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3455 014254 001012      BNE      72$      ;BR IF NOT
3456 014256 012737 177777 001254 71$:      MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
3457 014264 012760 000011 000000      MOV      #11, RMCS1(RO) ;CLEAR THE DRIVE
3458 014272 012760 000013 000000      MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
3459 014300 104026      ERROR    26      ;TYPE ERROR MESSAGE 26
3460 014302 013737 001170 001126 72$:      MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMOS1 READ
3461 014310 013737 001224 001240      MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
3462 014316 042737 100000 001126      BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
3463 014324 023737 001124 001126      CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
3464 014332 001401      BEQ      73$      ;BR IF OK FROM PORT A.
3465 014334 104007      ERROR    7 ;REPORT ERROR
3466 014336 013737 001172 001126 73$:      MOV      $TMP3, $BDDAT ;CHECK RMOS1 FOR BIT FAILURES - FROM PORT B.
3467 014344 013737 001226 001240      MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
3468 014352 042737 100000 001126      BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
3469 014360 023737 001124 001126      CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.

```

3470 014366 001401
3471 014370 104007
3472 014372 000240
3473 014374 000004

BEG 745 ;BR IF OK
ERROR 7 ;REPORT ERROR
745: NOP
15: SCOPE ;LOOP ?

```
*****
*TEST 11 PORT 'B' RELEASE INTERFERENCE TEST
*
*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
* IS SEIZED BY THE OTHER PORT.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
*
* C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
*
* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED
* TO PORT 'B'.
*
* E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED
* TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
```

3491 014376
3495 014376 005737 001300
3496 014402 001406
3497 014404 100002
3498 014406 000137 002676
3499 014412 012737 177777 001300
3500 014420 112737 000011 001102
3501 014426 012737 014450 001106
3502 014434 012737 014450 001110
3503 014442 012737 000031 001176
3504 014450 012706 001100
3505
3506
3507
3508 014454 113760 001224 000010
3509 014462 005060 000012
3510 014466 012760 000011 000000
3511 014474 012760 000013 000000
3512 014502 113760 001226 000010
3513 014510 005060 000012
3514 014514 012760 000011 000000
3515 014522 012760 000013 000000
3516
3517
3518
3519
3520
3521 014530 113760 001224 000010
3522 014536 013737 001224 001242
3523 014544 005060 000012
3524 014550 113760 001226 000010
3525 014556 013737 001226 001240

```
*****
*ST11:
* TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
* BEG 25 ;BR IF NOT
* BPL 15 ;BR IF JUST ENTERED TEST
* JMP EXEC ;RETURN & GET NEXT TEST NUMBER
* 15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
* 25: MOVB #11,$STNM ;TEST NUMBER
* MOV #TEST11,$LPADR ;LOAD LOOP ON TEST ADDRESS
* MOV #TEST11,$LPERR ;LOAD LOOP ON ERROR ADDRESS
* MOV #25,$TIMES ;DO 25 ITERATIONS
* TEST11: MOV #STACK,SP ;LOAD THE STACK POINTER
*
* ;CLEAR ATTENTION BITS FOR BOTH PORTS
*
* MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
* CLR RMDS1(RO) ;SEIZE THE DRIVE
* MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
* MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
* MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
* CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
* MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
* MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
*
* ;*****
*
* ;SEIZE THE DRIVE THROUGH PORT A
*
* MOVB PORTA,RMCS2(RO) ;SELECT PORT A
* MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
* CLR RMDS1(RO) ;WRITE RMDS1
* MOVB PORTB,RMCS2(RO) ;SELECT PORT B
* MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```

3526 014564 013737 001226 001244 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
3527 014572 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
3528 014600 010037 001122 MOV RO, $BDADR ;RH11 BASE ADDRESS
3529 014604 062737 000012 001122 ADD #RMDS1, $BDADR ;GENERATE BAD REGISTER ADDRESS
3530 014612 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
3531 014616 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
3532 014624 001403 BEQ 64$ ;BR IF IT IS
3533 014626 104004 ERROR 4 ;REPORT THE ERROR
3534 014630 000137 015640 JMP 1$ ;BYPASS REST OF THE SUBTEST
3535 014634 64$:
3536 014634 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
3537 014642 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3538 014650 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
3539 014656 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
3540 014664 012737 011700 001124 MOV #MOL!PGA!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
3541 014672 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
3542 014700 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
3543 014704 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
3544 014712 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
3545 014720 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
3546 014726 001401 BEQ 65$ ;BR IF THEY ARE
3547 014730 104005 ERROR 5 ;REPORT THE ERROR
3548 014732 000240 65$:
3549 NOP
3550
3551 ;:*****
3552 ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
3553 014734 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
3554 014742 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3555 014750 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT B
3556
3557 ;:*****
3558 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
3559
3560 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3561 014762 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
3562 014770 012737 000012 001122 MOV #RMDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3563 014776 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
3564 015002 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
3565 015006 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
3566 015014 001403 BEQ 66$ ;BR IF OK
3567 015016 104010 ERROR 10 ;REPORT THE ERROR
3568 015020 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3569 015024 016037 000000 001126 66$: MOV RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RMCS1
3570 015032 012737 000000 001122 MOV #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
3571 015040 060037 001122 ADD RO, $BDADR ;ADDRESS BASE
3572 015044 032737 020000 001126 BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
3573 015052 001404 BEQ 67$ ;BR IF NOT
3574 015054 104011 ERROR 11 ;REPORT THE ERROR
3575 015056 012760 040000 000000 MOV #TRE, RMCS1(RO) ;CLEAR 'MCPE'
3576 015064 000240 67$:
3577 015066 005737 001250 NOP
3578 015072 001402 TST CKERR ;WAS RMDS1 NON ZERO ?
3579 015074 000137 015640 BEQ .+6 ;CONTENTS OF RMDS1 SEEN BY PORT B
3580 JMP 1$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
3581
;:*****

```

E06

CZRMGB0 RM03-2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24MACY11 30(1046)
T1121-NOV-77 13:53 PAGE 69
PORT 'B' RELEASE INTERFERENCE TEST

SEQ 0069

```

3582                                     ;RELEASE THE DRIVE FROM PORT A
3583
3584
3585 015100 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3586 015106 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3587 015114 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
3588
3589                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
3590
3591 015122 005037 001254                CLR    RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
3592 015126 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3593 015134 012737 000012 001122      MOV    #RMS1, $BDADR ;REGISTER ADDRESS INCREMENT
3594 015142 060037 001122                ADD    RO, $BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
3595 015146 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3596 015154 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3597 015162 016037 000012 001164      MOV    RMS1(RO), $TMP0 ;READ STATUS REGISTER FROM PORT B
3598 015170 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3599 015176 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3600 015204 016037 000012 001126      MOV    RMS1(RO), $BDAT ;DRIVE STATUS FROM PORT A
3601 015212 001404                BEQ    68$ ;BR IF STATUS FROM PORT A ZERO
3602 015214 005737 001164                TST    $TMP0 ;IS STATUS FROM PORT B ZERO ?
3603 015220 001401                BEQ    68$ ;BR IF ZERO
3604 015222 104031                ERROR  31 ;REPORT DRIVE IN NEUTRAL
3605 015224 013737 001164 001126 68$: MOV    $TMP0, $BDAT ;CHECK STATUS FROM PORT B
3606 015232 013737 001226 001240      MOV    PORTB, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
3607 015240 023737 001124 001126      CMP    $GDDAT, $BDAT ;COMPARE WITH CONSTANT
3608 015246 001401                BEQ    69$ ;BR IF OK
3609 015250 104027                ERROR  27 ;REPORT REGISTER ERROR
3610 015252 000240                NOP
3611
3612                                     ;RELEASE THE DRIVE FROM PORT B
3613
3614 015254 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3615 015262 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3616 015270 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3617
3618                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3619
3620 015276 005037 001254                CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3621 015302 012737 000012 001122      MOV    #RMS1, $BDADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
3622 015310 060037 001122                ADD    RO, $BDADR ;ADD THE I/O BASE ADDRESS
3623 015314 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3624 015322 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
3625 015330 016037 000012 001170      MOV    RMS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3626 015336 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
3627 015344 013737 001170 001164      MOV    $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
3628 015352 042737 100100 001164      BIC    #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3629 015360 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
3630 015366 016037 000012 001172      MOV    RMS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3631 015374 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
3632 015402 013737 001172 001166      MOV    $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
3633 015410 042737 100100 001166      BIC    #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3634 015416 023737 001164 001166      CMP    $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3635 015424 001006                BNE    70$ ;BR IF NOT
3636 015426 005737 001164                TST    $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3637 015432 001045                BNE    72$ ;BR IF NOT

```

```

3638 015434 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3639 015436 000137 015636    JMP 74$          ;BYPASS THE REST OF THE CHECKS
3640 015442 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3641 015457 013737 001226 001240    MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3642 015456 113760 001226 000010    MOVVB PORTB,RMCS2(RO) ;SELECT PORT B.
3643 015464 005737 001164          TST $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
3644 015470 001414          BEQ 71$         ;BR IF ZERO
3645 015472 013737 001224 001240    MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3646 015500 013737 001172 001126    MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3647 015506 113760 001224 000010    MOVVB PORTA,RMCS2(RO) ;SELECT PORT A.
3648 015514 005737 001166          TST $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
3649 015520 001012          BNE 72$         ;BR IF NOT
3650 015522 012737 177777 001254 71$: MOV #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
3651 015530 012760 000011 000000    MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
3652 015536 012760 000013 000000    MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
3653 015544 104026          ERROR 26         ;TYPE ERROR MESSAGE 26
3654 015546 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3655 015554 013737 001224 001240    MOV PORTA,PTNBR ;CHANGE PORT NUMBER
3656 015562 042737 100000 001126    BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3657 015570 023737 001124 001126    CMP $GDDAT,$BDDAT ;ALL BITS OK ?
3658 015575 001401          BEQ 73$         ;BR IF OK FROM PORT A.
3659 015600 104007          ERROR 7          ;REPORT ERROR
3660 015602 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3661 015610 013737 001226 001240    MOV PORTB,PTNBR ;CHANGE PORT NUMBER
3662 015616 042737 100000 001126    BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3663 015624 023737 001124 001126    CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
3664 015632 001401          BEQ 74$         ;BR IF OK
3665 015634 104007          ERROR 7          ;REPORT ERROR
3666 015636 000240          NOP
3667 015640 000004          IS: SCOPE
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693

```

```

*****
*TEST 12 PORT 'A' RELEASE W/ERRORS TEST
*
*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
*BITS ARE SET IN THE DRIVE.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
*
* B. WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
*
* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO'
* BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
* THAT RMER1 HAS NOT BEEN CLEARED.
*
* D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
*
* E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****
*ST12:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ^
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST

```


G06

CZRMGB0 RMO3 2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 71
T12 PORT 'A' RELEASE W/ERRORS TEST

SEQ 0071

```

3694 015652 000137 002676      JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
3695 015656 012737 177777      001300 1$:      MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
3696 015664 112737 000012      001102 2$:      MOV      #12,$STNM     ;TEST NUMBER
3697 015672 012737 015714      001106      MOV      #TEST12,$LPADR ;LOAD LOOP ON TEST ADDRESS
3698 015700 012737 015714      001110      MOV      #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3699 015706 012737 000031      001176      MOV      #25,$TIMES    ;DO 25. ITERATIONS
3700 015714 012706 001100      TEST12: MOV     #STACK,SP   ;LOAD THE STACK POINTER
3701
3702      ;CLEAR ATTENTION BITS FOR BOTH PORTS
3703
3704 015720 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT #A
3705 015726 005060 000012      CLR      RMDS1(RO)      ;SEIZE THE DRIVE
3706 015732 012760 000011 000000      MOV      #11, RMCS1(RO)  ;ISSUE DRIVE CLEAR
3707 015740 012760 000013 000000      MOV      #13, RMCS1(RO)  ;RELEASE THE DRIVE
3708 015746 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT #B
3709 015754 005060 000012      CLR      RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
3710 015760 012760 000011 000000      MOV      #11, RMCS1(RO)  ;ISSUE DRIVE CLEAR
3711 015766 012760 000013 000000      MOV      #13, RMCS1(RO)  ;RELEASE THE DRIVE
3712      ;:*****
3713
3714      ;SEIZE THE DRIVE THROUGH PORT A
3715
3716 015774 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
3717 016002 013737 001224 001242      MOV      PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
3718 016010 005060 000012      CLR      RMDS1(RO)      ;WRITE RMDS1
3719 016014 013737 001226 001244      MOV      PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
3720
3721      ;:*****
3722      ;FORCE AN ERROR
3723
3724 016022 012760 177777 000014      MOV      #-1, RMER1(RO)  ;SET ERROR BITS
3725 016030 012760 000013 000000      MOV      #13, RMCS1(RO)  ;ISSUE A RELEASE COMMAND
3726 016036 005037 001250      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
3727 016042 016037 000000 001126      MOV      RMCS1(RO), $BDDAT ;GET CONTENTS OF RMCS1
3728 016050 012737 000000 001122      MOV      #RMCS1, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3729 016056 060037 001122      ADD      RO, $B0ADR      ;ADD RH11 BASE ADDRESS
3730 016062 012737 004012 001124      MOV      #4012, $GDDAT   ;WHAT REGISTER SHOULD BE
3731 016070 013737 001126 001164      MOV      $GDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
3732 016076 042737 173765 001164      BIC      #4012, $TMP0    ;SAVE SPECIFIED BITS
3733 016104 023737 001124 001164      CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
3734 016112 001414      BEQ      66$            ;BR IF OK
3735 016114 013737 001126 001174      MOV      $B0LAT, $TMP4   ;COPY 'BAD DATA'
3736 016122 042737 004012 001174      BIC      #4012, $TMP4    ;CLEAR THE MASKED BITS
3737 016130 053737 001174 001124      BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
3738 016136 104025      ERROR    25            ;TYPE MESSAGE 25
3739 016140 005137 001250      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
3740 016144 000240      66$:      NOP
3741 016146 005737 001250      TST     CKERR          ;DID 'GO' BIT RESET ?
3742 016152 001002      BNE     +6            ;BR IF NOT
3743 016154 000137 016214      JMP     1$            ;'GO' BIT RESET
3744 016160 012760 000040 000010      MOV      #CLR, RMCS2(RO) ;INIT THE RH11
3745 016166 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
3746 016174 013737 001224 001240      MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3747 016202 012760 000013 000000      MOV      #13, RMCS1(RO)  ;RELEASE THE DRIVE THROUGH PORT A
3748 016210 000137 016760      JMP     2$            ;BYPASS THE REST OF THE TEST
3749

```


H06

CZRMGB0 RMO3 2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 72
T12 PORT 'A' RELEASE W/ERRORS TEST

SEQ 0072

```

3750
3751
3752
3753 016214
3754 016214 113760 001226 000010
3755 016222 013737 001226 001240
3756 016230 005037 001250
3757 016234 016037 000012 001126
3758 016242 012737 000012 001122
3759 016250 060037 001122
3760 016254 005037 001124
3761 016260 023737 001124 001126
3762 016266 001403
3763 016270 104024
3764 016272 005137 001250
3765 016276 000240
3766 016300 113760 001224 000010
3767 016306 013737 001224 001240
3768 016314 005037 001250
3769 016320 016037 000014 001126
3770 016326 012737 000014 001122
3771 016334 060037 001122
3772 016340 012737 177777 001124
3773 016346 023737 001124 001126
3774 016354 001403
3775 016356 104010
3776 016360 005137 001250
3777 016364 000240
3778
3779
3780
3781
3782 016366 012760 000011 000000
3783
3784
3785
3786
3787
3788 016374 113760 001224 000010
3789 016402 013737 001224 001240
3790 016410 012760 000013 000000
3791
3792
3793
3794 016416 005037 001254
3795 016422 012737 000012 001122
3796 016430 060037 001122
3797 016434 012737 011700 001124
3798 016442 113760 001224 000010
3799 016450 016037 000012 001170
3800 016456 042737 024001 001170
3801 016464 013737 001170 001164
3802 016472 042737 100100 001164
3803 016500 113760 001226 000010
3804 016506 016037 000012 001172
3805 016514 042737 024001 001172

;*****
;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
15:
MOV B PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
MOV #RMDS1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAADR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
BEQ 68$ ;BR IF OK
ERROR 24 ;TYPE MESSAGE 24
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
68$:
NOP
MOV B PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
MOV #RMER1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAADR ;ADD RH11 BASE ADDRESS
MOV #17777, $GDDAT ;WHAT REGISTER SHOULD BE
CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
BEQ 70$ ;BR IF OK
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
70$:
NOP

;*****
;CLEAR THE ERRORS THROUGH PORT A
MOV #11, RMCS1(RO) ;ISSUE A DRIVE CLEAR

;*****
;RELEASE THE DRIVE FROM PORT A
MOV B PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RMDS1, $BDAADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
ADD RO, $BDAADR ;ADD THE I/O BASE ADDRESS
MOV #MOL:PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
MOV B PORTA, RMCS2(RO) ;SELECT PORT A
MOV RMDS1(RO), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
BIC #PIP!WRL!OM, $STMP2 ;CLEAR DONT CARES
MOV $STMP2, $STMP0 ;COPY IT INTO '$STMP0'
BIC #ATA!VV, $STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOV B PORTB, RMCS2(RO) ;SELECT PORT B
MOV RMDS1(RO), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
BIC #PIP!WRL!OM, $STMP3 ;CLEAR DONT CARES

```

```

3806 016522 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
3807 016530 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3808 016536 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3809 016544 001006 BNE 72$ ;BR IF NOT
3810 016546 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3811 016552 001045 BNE 74$ ;BR IF NOT
3812 016554 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3813 016556 000137 016756 JMP 76$ ;BYPASS THE REST OF THE CHECKS
3814 016562 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3815 016570 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3816 016576 113760 001226 000010 MOVB PORTB,AMCS2(RO) ;SELECT PORT B.
3817 016604 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3818 016610 001414 BEQ 73$ ;BR IF ZERO
3819 016612 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3820 016620 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3821 016626 113760 001224 000010 MOVB PORTA,AMCS2(RO) ;SELECT PORT A.
3822 016634 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3823 016640 001012 BNE 74$ ;BR IF NOT
3824 016642 012737 177777 001254 73$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
3825 016650 012760 000011 000000 MOV #11,AMCS1(RO) ;CLEAR THE DRIVE
3826 016656 012760 000013 000000 MOV #13,AMCS1(RO) ;RELEASE THE DRIVE
3827 016664 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
3828 016666 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
3829 016674 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
3830 016702 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3831 016710 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
3832 016716 001401 BEQ 75$ ;BR IF OK FROM PORT A.
3833 016720 104007 ERROR 7 ;REPORT ERROR
3834 016722 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
3835 016730 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
3836 016736 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
3837 016744 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
3838 016752 001401 BEQ 76$ ;BR IF OK
3839 016754 104007 ERROR 7 ;REPORT ERROR
3840 016756 000240 NOP
3841 016760 000004 76$: SCOPE ;LOOP ?

```

```

3842
3843 *****
3844 *TEST 13 PORT 'B' RELEASE W/ERRORS TEST
3845
3846 *VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
3847 *BITS ARE SET IN THE DRIVE.
3848
3849 * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
3850
3851 * B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
3852
3853 * C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO'
3854 * BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
3855 * THAT RMER1 HAS NOT BEEN CLEARED.
3856
3857 * D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
3858
3859 * E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
3860 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3861

```

```

3862
3863 016762
3864 016762 005737 001300
3865 016766 001406
3866 016770 100002
3867 016772 000137 002676
3868 016776 012737 177777 001300
3869 017004 112737 000013 001102
3870 017012 012737 017034 001106
3871 017020 012737 017034 001110
3872 017026 012737 000031 001176
3873 017034 012706 001100
3874
3875
3876
3877 017040 113760 001224 000010
3878 017046 005060 000012
3879 017052 012760 000011 000000
3880 017060 012760 000013 000000
3881 017066 113760 001226 000010
3882 017074 005060 000012
3883 017100 012760 000011 000000
3884 017106 012760 000013 000000
3885
3886
3887
3888
3889 017114 113760 001226 000010
3890 017122 013737 001226 001242
3891 017130 005060 000012
3892 017134 013737 001224 001244
3893
3894
3895
3896
3897 017142 012760 177777 000014
3898 017150 012760 000013 000000
3899 017156 005037 001250
3900 017162 016037 000000 001126
3901 017170 012737 000000 001122
3902 017176 060037 001122
3903 017202 012737 004012 001124
3904 017210 013737 001126 001164
3905 017216 042737 173765 001164
3906 017224 023737 001124 001164
3907 017232 001414
3908 017234 013737 001126 001174
3909 017242 042737 004012 001174
3910 017250 053737 001174 001124
3911 017256 104025
3912 017260 005137 001250
3913 017264 000240
3914 017266 005737 001250
3915 017272 001002
3916 017274 000137 017334
3917 017300 012760 000040 000010

```

```

;*****
TST13: 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 #13,$STSTM ;TEST NUMBER
      MOV #TEST13,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV #25,$TIMES ;DO 25 ITERATIONS
TEST13: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

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

;FORCE AN ERROR
MOV #-1, RMER1(RO) ;SET ERROR BITS
MOV #13, RMCS1(RO) ;ISSUE A RELEASE COMMAND
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMCS1(RO), $BDDAT ;GET CONTENTS OF RMCS1
MOV #RMCS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
MOV #4012, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMPD ;MOVE REGISTER CONTENTS TO 'TMPD'
BIC #4012, $TMPD ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMPD ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #4012, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 25 ;TYPE MESSAGE 25
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
66$: NOP
      TST CKERR ;DID 'GO' BIT RESET ?
      BNE .+6 ;BR IF NOT
      JMP 15 ;'GO' BIT RESET
      MOV #CLR, RMCS2(RO) ;INIT THE RH11

```

```

3918 017306 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3919 017314 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3920 017322 012760 000013 000000      MOV    #13, RMCS1(RO) ;RELEASE THE DRIVE THROUGH PORT B
3921 017330 000137 020100      JMP    25 ;BYPASS THE REST OF THE TEST
3922
3923 ;*****
3924 ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
3925
3926 017334      1S:
3927 017334 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
3928 017342 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3929 017350 005007 001250      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3930 017354 016037 000012 001126      MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
3931 017362 012737 000012 001122      MOV    #RMDS1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3932 017370 060037 001122      ADD    RO, $BDAADR ;ADD RH11 BASE ADDRESS
3933 017374 005037 001124      CLR    $GDDAT ;WHAT REGISTER SHOULD BE
3934 017400 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
3935 017406 001403      BEQ    68$ ;BR IF OK
3936 017410 104024      ERROR 24 ;TYPE MESSAGE 24
3937 017412 005137 001250      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3938 017416 000240      NOP
3939 017420 113760 001226 000010      68$: MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3940 017426 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3941 017434 005037 001250      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3942 017440 016037 000014 001126      MOV    RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
3943 017446 012737 000014 001122      MOV    #RMER1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3944 017454 060037 001122      ADD    RO, $BDAADR ;ADD RH11 BASE ADDRESS
3945 017460 012737 177777 001124      MOV    #177777, $GDDAT ;WHAT REGISTER SHOULD BE
3946 017466 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
3947 017474 001403      BEQ    70$ ;BR IF OK
3948 017476 104010      ERROR 10 ;REPORT THE ERROR
3949 017500 005137 001250      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3950 017504 000240      NOP
3951
3952 ;*****
3953 ;CLEAR THE ERRORS THROUGH PORT B
3954
3955 017506 012760 000011 000000      MOV    #11, RMCS1(RO) ;ISSUE A DRIVE CLEAR
3956
3957 ;*****
3958 ;RELEASE THE DRIVE FROM PORT B
3959
3960
3961 017514 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
3962 017522 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3963 017530 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
3964
3965 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3966
3967 017536 005037 001254      CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3968 017542 012737 000012 001122      MOV    #RMDS1, $BDAADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
3969 017550 060037 001122      ADD    RO, $BDAADR ;ADD THE I/O BASE ADDRESS
3970 017554 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
3971 017562 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
3972 017570 016037 000012 001170      MOV    RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3973 017576 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES

```

```

3974 017604 013737 001170 001164      MOV      $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
3975 017612 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3976 017620 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
3977 017626 016037 000012 001172      MOV      RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3978 017634 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
3979 017642 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3980 017650 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3981 017656 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3982 017664 001006      BNE      72$             ;BR IF NOT
3983 017666 005737 001164      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3984 017672 001045      BNE      74$             ;BR IF NOT
3985 017674 104046      ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3986 017676 000137 020076      JMP      76$            ;BYPASS THE REST OF THE CHECKS
3987 017702 013737 001170 001126 72$:      MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3988 017710 013737 001226 001240      MOV      PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3989 017716 113760 001226 000010      MOV      PORTB, RMCS2(RO);SELECT PORT B.
3990 017724 005737 001164      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3991 017730 001414      BEQ      73$            ;BR IF ZERO
3992 017732 013737 001224 001240      MOV      PORTA, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3993 017740 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3994 017746 113760 001224 000010      MOV      PORTA, RMCS2(RO);SELECT PORT A.
3995 017754 005737 001166      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3996 017760 001012      BNE      74$            ;BR IF NOT
3997 017762 012737 177777 001254 73$:      MOV      #-1, RELERR     ;SET 'RELEASE ERROR' INDICATOR
3998 017770 012760 000011 000000      MOV      #11, RMCS1(RO)  ;CLEAR THE DRIVE
3999 017776 012760 000013 000000      MOV      #13, RMCS1(RO)  ;RELEASE THE DRIVE
4000 020004 104026      ERROR    26             ;TYPE ERROR MESSAGE 26
4001 020006 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4002 020014 013737 001224 001240      MOV      PORTA, PTNBR    ;CHANGE PORT NUMBER
4003 020022 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
4004 020030 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
4005 020036 001401      BEQ      75$            ;BR IF OK FROM PORT A.
4006 020040 104007      ERROR    7             ;REPORT ERROR
4007 020042 013737 001172 001126 75$:      MOV      $TMP3,$BDDAT    ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4008 020050 013737 001226 001240      MOV      PORTB, PTNBR    ;CHANGE PORT NUMBER
4009 020056 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
4010 020064 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
4011 020072 001401      BEQ      76$            ;BR IF OK
4012 020074 104007      ERROR    7             ;REPORT ERROR
4013 020076 000240      NOP
4014 020100 000004      2$:      SCOPE                  ;LOOP ?

```

```

4015
4016
4017
4018 *****
4019 *TEST 14      PORT 'A' SEIZE AND CLEAR TEST
4020 *
4021 *VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
4022 *      PORT TO RELEASE THE DRIVE.
4023 *
4024 * A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS1 THROUGH PORT 'A'.
4025 *      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
4026 *
4027 * B. ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
4028 *      DOES NOT RETURN TO NEUTRAL.
4029 *
4030 * C. ISSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE

```

M06

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 77
T14 PORT 'A' SEIZE AND CLEAR TEST

SEQ 0077

```

4030 ;* DOES NOT RETURN TO NEUTRAL.
4031 ;*
4032 ;* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
4033 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4034 ;*
4035 ;*****
4036 TST14: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
4037 020102 005737 001300 BEQ 25 ;BR IF NOT
4038 020106 001406 BPL 15 ;BR IF JUST ENTERED TEST
4039 020110 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
4040 020112 000137 002676 15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
4041 020116 012737 177777 001300 25: MOV #14,$STNM ;TEST NUMBER
4042 020124 112737 000014 001102 MOV #TEST14,$LPADR ;LOAD LOOP ON TEST ADDRESS
4043 020132 012737 020154 001106 MOV #TEST14,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4044 020140 012737 020154 001110 MOV #25,$TIMES ;DO 25. ITERATIONS
4045 020146 012737 000031 001176 TEST14: MOV #STACK,SP ;LOAD THE STACK POINTER
4046 020154 012706 001100
4047
4048 ;;*****
4049 ;SEIZE THE DRIVE THROUGH PORT A
4050
4051 MOV# PORTA, RMCS2(RO) ;SELECT PORT A
4052 020160 113760 001224 000010 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
4053 020166 013737 001224 001242 CLR RMDS1(RO) ;WRITE RMDS1
4054 020174 005060 000012 MOV# PORTB, RMCS2(RO) ;SELECT PORT B
4055 020200 113760 001226 000010 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4056 020206 013737 001226 001240 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
4057 020214 013737 001226 001244 MOV RMDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
4058 020222 016037 000012 001126 MOV RO, $BDAADR ;R#11 BASE ADDRESS
4059 020230 010037 001122 ADD #RMDS1, $BDAADR ;GENERATE BAD REGISTER ADDRESS
4060 020234 062737 000012 001122 CLR $GDDAT ;REGISTER SHOULD BE ZERO
4061 020242 005037 001124 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
4062 020246 023737 001124 001126 BEQ 645 ;BR IF IT IS
4063 020254 001403 ERROR 4 ;REPORT THE ERROR
4064 020256 104004 JMP 15 ;BYPASS REST OF THE SUBTEST
4065 020260 000137 021500
4066 020264
4067 020264 113760 001224 000010 645: MOV# PORTA, RMCS2(RO) ;SELECT PORT A
4068 020272 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4069 020300 016037 000012 001126 MOV RMCS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4070 020306 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
4071 020314 012737 011700 001124 MOV #MOL!PGH!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
4072 020322 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
4073 020330 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
4074 020334 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
4075 020342 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
4076 020350 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4077 020356 001401 BEQ 655 ;BR IF THEY ARE
4078 020360 104005 ERROR 5 ;REPORT THE ERROR
4079 020362 000240 655: NOP
4080
4081 ;;*****
4082 ;DRIVE CLEAR THROUGH PORT A FIRST
4083
4084 020364 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT A
4085

```

```

4086
4087
4088
4089 020372 113760 001226 000010      MOV  PORTB, RMCS2(RO) ; SELECT PORT B
4090 020400 013737 001226 001240      MOV  PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4091 020406 005037 001250      CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
4092 020412 016037 000012 001126      MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
4093 020420 012737 000012 001122      MOV  #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
4094 020426 060037 001122      ADD  RO, $BDAOR ; ADD RH11 BASE ADDRESS
4095 020432 005037 001124      CLR  $GDDAT ; WHAT REGISTER SHOULD BE
4096 020436 013737 001126 001164      MOV  $BDDAT, $TMPD ; MOVE REGISTER CONTENTS TO '$TMPD'
4097 020444 042737 100000 001164      BIC  #1C7777, $TMPD ; SAVE SPECIFIED BITS
4098 020452 023737 001124 001164      CMP  $GDDAT, $TMPD ; COMPARE THE BITS
4099 020460 001414      BEQ  66$ ; BR IF OK
4100 020462 013737 001126 001174      MOV  $BDDAT, $TMP4 ; COPY 'BAD DATA'
4101 020470 042737 077777 001174      BIC  #77777, $TMP4 ; CLEAR THE MASKED BITS
4102 020476 053737 001174 001124      BIS  $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
4103 020504 104033      ERROR 33 ; TYPE MESSAGE 33
4104 020506 005137 001250      COM  CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
4105 020512 000240      NOP
4106 020514 113760 001224 000010      MOV  PORTA, RMCS2(RO) ; SELECT PORT A
4107 020522 013737 001224 001240      MOV  PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4108 020530 005037 001250      CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
4109 020534 016037 000012 001126      MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
4110 020542 012737 000012 001122      MOV  #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
4111 020550 060037 001122      ADD  RO, $BDAOR ; ADD RH11 BASE ADDRESS
4112 020554 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ; WHAT REGISTER SHOULD BE
4113 020562 013737 001126 001164      MOV  $BDDAT, $TMPD ; MOVE REGISTER CONTENTS TO '$TMPD'
4114 020570 042737 100000 001164      BIC  #1C7777, $TMPD ; SAVE SPECIFIED BITS
4115 020576 023737 001124 001164      CMP  $GDDAT, $TMPD ; COMPARE THE BITS
4116 020604 001414      BEQ  68$ ; BR IF OK
4117 020606 013737 001126 001174      MOV  $BDDAT, $TMP4 ; COPY 'BAD DATA'
4118 020614 042737 077777 001174      BIC  #77777, $TMP4 ; CLEAR THE MASKED BITS
4119 020622 053737 001174 001124      BIS  $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
4120 020630 104033      ERROR 33 ; TYPE MESSAGE 33
4121 020632 005137 001250      COM  CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
4122 020636 000240      NOP
4123
4124
4125
4126
4127 020640 012760 000040 000010      MOV  #CLR, RMCS2(RO) ; ISSUE MASSBUS INIT
4128
4129
4130
4131
4132 020646 113760 001226 000010      MOV  PORTB, RMCS2(RO) ; SELECT PORT B
4133 020654 013737 001226 001240      MOV  PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4134 020662 005037 001250      CLR  CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
4135 020666 016037 000012 001126      MOV  RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
4136 020674 012737 000012 001122      MOV  #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
4137 020702 060037 001122      ADD  RO, $BDAOR ; ADD RH11 BASE ADDRESS
4138 020706 005037 001124      CLR  $GDDAT ; WHAT REGISTER SHOULD BE
4139 020712 013737 001126 001164      MOV  $BDDAT, $TMPD ; MOVE REGISTER CONTENTS TO '$TMPD'
4140 020720 042737 100000 001164      BIC  #1C7777, $TMPD ; SAVE SPECIFIED BITS
4141 020726 023737 001124 001164      CMP  $GDDAT, $TMPD ; COMPARE THE BITS

```

; VERIFY THAT DRIVE STILL SEIZED BY PORT A

66\$:

68\$:

; NOW ISSUE MASSBUS INIT

; CONFIRM THAT DRIVE STILL SEIZED BY PORT A


```

4142 020734 001414          BEQ      70$          ;BR IF OK
4143 020736 013737 001126 001174  MOV     $B00AT,$STMP4 ;COPY 'BAD DATA'
4144 020744 042737 077777 001174  BIC     #77777,$STMP4 ;CLEAR THE MASKED BITS
4145 020752 053737 001174 001124  BIS     $STMP4,$G00AT ;'OR' WITH GOOD DATA FOR TYPEOUT
4146 020760 104034          ERROR    34          ;TYPE MESSAGE 34
4147 020762 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
4148 020766 000240          NOP
4149 020770 113760 001224 000010 70$:  MOV     PORTA, RMCS2(RO) ;SELECT PORT A
4150 020776 013737 001224 001240  MOV     PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4151 021004 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
4152 021010 016037 000012 001126  MOV     RMDS1(RO), $B00AT ;GET CONTENTS OF RMDS1
4153 021016 013737 000012 001122  MOV     #RMDS1, $B00ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4154 021024 060037 001122          ADD     RO, $B00ADR  ;ADD RHI1 BASE ADDRESS
4155 021030 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$G00AT ;WHAT REGISTER SHOULD BE
4156 021036 013737 001126 001164  MOV     $B00AT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4157 021044 042737 100000 001164  BIC     #1C7777,$STMP0 ;SAVE SPECIFIED BITS
4158 021052 023737 001124 001164  CMP     $G00AT,$STMP0 ;COMPARE THE BITS
4159 021060 001414          BEQ     72$          ;BR IF OK
4160 021062 013737 001126 001174  MOV     $B00AT,$STMP4 ;COPY 'BAD DATA'
4161 021070 042737 077777 001174  BIC     #77777,$STMP4 ;CLEAR THE MASKED BITS
4162 021076 053737 001174 001124  BIS     $STMP4,$G00AT ;'OR' WITH GOOD DATA FOR TYPEOUT
4163 021104 104034          ERROR    34          ;TYPE MESSAGE 34
4164 021106 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
4165 021112 000240          NOP
4166
4167
4168          ;RELEASE THE DRIVE FROM PORT A
4169 021114 113760 001224 000010  MOV     PORTA, RMCS2(RO) ;SELECT PORT A
4170 021122 013737 001224 001240  MOV     PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4171 021130 012760 000013 000000  MOV     #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4172
4173          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4174
4175 021136 005037 001254          CLR     RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
4176 021142 012737 000012 001122  MOV     #RMDS1, $B00ADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4177 021150 060037 001122          ADD     RO, $B00ADR  ;ADD THE I/O BASE ADDRESS
4178 021154 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$G00AT ;COMPARISON CONSTANT
4179 021162 113760 001224 000010  MOV     PORTA, RMCS2(RO) ;SELECT PORT A.
4180 021170 016037 000012 001170  MOV     RMDS1(RO), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4181 021176 042737 024001 001170  BIC     #PIP!WAL!OM,$STMP2 ;CLEAR DONT CARES
4182 021204 013737 001170 001164  MOV     $STMP2,$STMP0 ;COPY IT INTO 'STMP0'
4183 021212 042737 100100 001164  BIC     #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4184 021220 113760 001226 000010  MOV     PORTB, RMCS2(RO) ;SELECT PORT B.
4185 021226 016037 000012 001172  MOV     RMDS1(RO), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4186 021234 042737 024001 001172  BIC     #PIP!WAL!OM,$STMP3 ;CLEAR DONT CARES
4187 021242 013737 001172 001166  MOV     $STMP3,$STMP1 ;COPY IT INTO 'STMP1'
4188 021250 042737 100100 001166  BIC     #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4189 021256 023737 001164 001166  CMP     $STMP0,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4190 021264 001006          BNE     74$          ;BR IF NOT
4191 021266 005737 001164          TST     $STMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4192 021272 001045          BNE     76$          ;BR IF NOT
4193 021274 104046          ERROR    46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4194 021276 000137 021476          JMP     78$          ;BYPASS THE REST OF THE CHECKS
4195 021302 013737 001170 001126 74$:  MOV     $STMP2,$B00AT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4196 021310 013737 001226 001240  MOV     PORTB, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4197 021316 113760 001226 000010  MOV     PORTB, RMCS2(RO) ;SELECT PORT B.

```



```

4198 021324 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4199 021330 001414 BEQ 75$ ;BR IF ZERO
4200 021332 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4201 021340 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4202 021346 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A.
4203 021354 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4204 021360 001012 BNE 76$ ;BR IF NOT
4205 021362 012737 177777 001254 75$: MOV #-1,FELERR ;SET 'RELEASE ERROR' INDICATOR
4206 021370 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
4207 021376 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
4208 021404 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
4209 021406 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4210 021414 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
4211 021422 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4212 021430 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
4213 021436 001401 BEQ 77$ ;BR IF OK FROM PORT A.
4214 021440 104007 ERROR 7 ;REPORT ERROR
4215 021442 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4216 021450 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4217 021456 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4218 021464 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4219 021472 001401 BEQ 78$ ;BR IF OK
4220 021474 104007 ERROR 7 ;REPORT ERROR
4221 021476 000240 78$: NOP
4222 021500 000004 1$: SCOPE ;LOOP ?

```

```

*****
*TEST 15 PORT 'B' SEIZE AND CLEAR TEST
*****
*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
* PORT TO RELEASE THE DRIVE.
*
* A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS1 THROUGH PORT 'B'.
* VERIFY THAT THE DRIVE HAS BEEN SEIZED.
*
* B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
* DOES NOT RETURN TO NEUTRAL.
*
* C. ISSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE
* DOES NOT RETURN TO NEUTRAL.
*
* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****

```

```

4243 021502 TST15: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
4244 021502 005737 001300 BEQ 2$ ;BR IF NOT
4245 021506 001406 BPL 1$ ;BR IF JUST ENTERED TEST
4246 021510 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
4247 021512 000137 002676 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
4248 021516 012737 177777 001300 2$: MOVVB #15,$STSTM ;TEST NUMBER
4249 021524 112737 000015 001102 MOV #TEST15,$LPADR ;LOAD LOOP ON TEST ADDRESS
4250 021532 012737 021554 001106 MOV #TEST15,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4251 021540 012737 021554 001110 MOV #25,$TIMES ;DO 25 ITERATIONS
4252 021546 012737 000031 001176 TEST15: MOV #STACK,SP ;LOAD THE STACK POINTER
4253 021554 012706 001100

```

```

4254
4255 ;*****
4256
4257 ;SEIZE THE DRIVE THROUGH PORT B
4258
4259 021560 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4260 021566 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
4261 021574 005060 000012 CLR RMDS1(RO) ;WRITE RMDS1
4262 021600 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4263 021606 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4264 021614 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
4265 021622 016037 000012 MOV RMDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
4266 021630 010037 001122 MOV RO, $BDAOR ;RH11 BASE ADDRESS
4267 021634 062737 000012 001122 ADD #RMDS1, $BDAOR ;GENERATE BAD REGISTER ADDRESS
4268 021642 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
4269 021646 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
4270 021654 001403 BEQ 64$ ;BR IF IT IS
4271 021656 104004 ERROR 4 ;REPORT THE ERROR
4272 021660 000137 023100 JMP 1$ ;BYPASS REST OF THE SUBTEST
4273 021664
4274 021664 113760 001226 000010 64$: MOVB PORTB, RMCS2(RO) ;SELECT PORT B
4275 021672 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4276 021700 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4277 021706 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
4278 021714 012737 011700 001124 MOV #MOL!PGH!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
4279 021722 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
4280 021730 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
4281 021734 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
4282 021742 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
4283 021750 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ^
4284 021756 001401 BEQ 65$ ;BR IF THEY ARE
4285 021760 104005 ERROR 5 ;REPORT THE ERROR
4286 021762 000240 65$: NOP
4287
4288 ;*****
4289 ;DRIVE CLEAR THROUGH PORT B FIRST
4290
4291 021764 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT B
4292
4293 ;*****
4294 ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
4295
4296 021772 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
4297 022000 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4298 022006 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4299 022012 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
4300 022020 012737 000012 001122 MOV #RMDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4301 022026 060037 001122 ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
4302 022032 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
4303 022036 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4304 022044 042737 100000 001164 BIC #1C7777, $TMP0 ;SAVE SPECIFIED BITS
4305 022052 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
4306 022060 001414 BEQ 66$ ;BR IF OK
4307 022062 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
4308 022070 042737 077777 001174 BIC #77777, $TMP4 ;CLEAR THE MASKED BITS
4309 022076 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

E07

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

IACY11 30(1046) 21-NOV-77 13:53 PAGE 82
T15 PORT 'B' SEIZE AND CLEAR TEST

SEG 0082

```

4310 022104 104033          ERROR 33          ;TYPE MESSAGE 33
4311 022106 005137 001250    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
4312 022112 000240          66$:  NOP
4313 022114 113760 001226 000010  MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
4314 022122 013737 001226 001240  MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4315 022130 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
4316 022134 016037 000012 001126  MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
4317 022142 012737 000012 001122  MOV    #RMDS1, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4318 022150 060037 001122          ADD    RO, $B0ADR     ;ADD RH11 BASE ADDRESS
4319 022154 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
4320 022162 013737 001126 001164  MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
4321 022170 042737 100000 001164  BIC    #1C7777, $TMP0  ;SAVE SPECIFIED BITS
4322 022176 023737 001124 001164  CMP    $GDDAT, $TMP0  ;COMPARE THE BITS
4323 022204 001414          BEQ    68$           ;BR IF OK
4324 022206 013737 001126 001174  MOV    $BDDAT, $TMP4   ;COPY 'BAD DATA'
4325 022214 042737 077777 001174  BIC    #77777, $TMP4   ;CLEAR THE MASKED BITS
4326 022222 053737 001174 001124  BIS    $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
4327 022230 104033          ERROR 33          ;TYPE MESSAGE 33
4328 022232 005137 001250    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
4329 022236 000240          68$:  NOP
4330
4331          ;*****
4332          ;NOW ISSUE MASSBUS INIT
4333
4334 022240 012760 000040 000010  MOV    #CLR, RMCS2(RO) ;ISSUE MASSBUS INIT
4335
4336          ;*****
4337          ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
4338
4339 022246 113760 001224 000010  MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
4340 022254 013737 001224 001240  MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4341 022262 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
4342 022266 016037 000012 001126  MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
4343 022274 012737 000012 001122  MOV    #RMDS1, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4344 022302 060037 001122          ADD    RO, $B0ADR     ;ADD RH11 BASE ADDRESS
4345 022306 005037 001124          CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
4346 022312 013737 001126 001164  MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
4347 022320 042737 100000 001164  BIC    #1C7777, $TMP0  ;SAVE SPECIFIED BITS
4348 022326 023737 001124 001164  CMP    $GDDAT, $TMP0  ;COMPARE THE BITS
4349 022334 001414          BEQ    70$           ;BR IF OK
4350 022336 013737 001126 001174  MOV    $BDDAT, $TMP4   ;COPY 'BAD DATA'
4351 022344 042737 077777 001174  BIC    #77777, $TMP4   ;CLEAR THE MASKED BITS
4352 022352 053737 001174 001124  BIS    $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
4353 022360 104034          ERROR 34          ;TYPE MESSAGE 34
4354 022362 005137 001250    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
4355 022366 000240          70$:  NOP
4356 022370 113760 001226 000010  MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
4357 022376 013737 001226 001240  MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4358 022404 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
4359 022410 016037 000012 001126  MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
4360 022416 012737 000012 001122  MOV    #RMDS1, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4361 022424 060037 001122          ADD    RO, $B0ADR     ;ADD RH11 BASE ADDRESS
4362 022430 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
4363 022436 013737 001126 001164  MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
4364 022444 042737 100000 001164  BIC    #1C7777, $TMP0  ;SAVE SPECIFIED BITS
4365 022452 023737 001124 001164  CMP    $GDDAT, $TMP0  ;COMPARE THE BITS

```

F07

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 83
T15 PORT 'B' SEIZE AND CLEAR TEST

SEQ 0083

4366	022460	001414				BEG	72\$;BR IF OK
4367	022462	013737	001126	001174		MOV	\$BDDAT,\$TMP4		;COPY 'BAD DATA'
4368	022470	042737	077777	001174		BIC	#77777,\$TMP4		;CLEAR THE MASKED BITS
4369	022476	053737	001174	001124		BIS	\$TMP4,\$GDDAT		; 'OR' WITH GOOD DATA FOR TYPEOUT
4370	022504	104034				ERROR	34		;TYPE MESSAGE 34
4371	022506	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
4372	022512	000240			72\$:	NOP			
4373									
4374									;RELEASE THE DRIVE FROM PORT B
4375									
4376	022514	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B
4377	022522	013737	001226	001240		MOV	PORTB, PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4378	022530	012760	000013	000000		MOV	#13, RMCS1(RO)		;ISSUE RELEASE THROUGH PORT B
4379									
4380									;VERIFY THAT THE DRIVE IS IN NEUTRAL
4381									
4382	022536	005037	001254			CLR	RELEA		;CLEAR THE 'RELEASE ERROR' INDICATOR
4383	022542	012737	000012	001122		MOV	#RMDS1,\$BDAOR		;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4384	022550	060037	001122			ADD	RO,\$BDAOR		;ADD THE I/O BASE ADDRESS
4385	022554	012737	011700	001124		MOV	#MOL:PGM:DPR:DRY:VV,\$GDDAT		;COMPARISON CONSTANT
4386	022562	113760	001224	000010		MOVB	PORTA, RMCS2(RO)		;SELECT PORT A.
4387	022570	016037	000012	001170		MOV	RMDS1(RO), \$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.
4388	022576	042737	024001	001170		BIC	#PIP:WRL:OM,\$TMP2		;CLEAR DONT CARES
4389	022604	013737	001170	001164		MOV	\$TMP2,\$TMP0		;COPY IT INTO 'TMP0'
4390	022612	042737	100100	001164		BIC	#ATA:VV,\$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4391	022620	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B.
4392	022626	016037	000012	001172		MOV	RMDS1(RO), \$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.
4393	022634	042737	024001	001172		BIC	#PIP:WRL:OM,\$TMP3		;CLEAR DONT CARES
4394	022642	013737	001172	001166		MOV	\$TMP3,\$TMP1		;COPY IT INTO 'TMP1'
4395	022650	042737	100100	001166		BIC	#ATA:VV,\$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4396	022656	023737	001164	001166		CMP	\$TMP0,\$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4397	022664	001006				BNE	74\$;BR IF NOT
4398	022666	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4399	022672	001045				BNE	76\$;BR IF NOT
4400	022674	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4401	022676	000137	023076			JMP	78\$;BYPASS THE REST OF THE CHECKS
4402	022702	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4403	022710	013737	001226	001240		MOV	PORTB, PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4404	022716	113760	001226	000010		MOVB	PORTB, RMCS2(RO)		;SELECT PORT B.
4405	022724	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
4406	022730	001414				BEG	75\$;BR IF ZERO
4407	022732	013737	001224	001240		MOV	PORTA, PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4408	022740	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
4409	022746	113760	001224	000010		MOVB	PORTA, RMCS2(RO)		;SELECT PORT A.
4410	022754	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
4411	022760	001012				BNE	76\$;BR IF NOT
4412	022762	012737	177777	001254	75\$:	MOV	#-1, RELEA		;SET 'RELEASE ERROR' INDICATOR
4413	022770	012760	000011	000000		MOV	#11, RMCS1(RO)		;CLEAR THE DRIVE
4414	022776	012760	000013	000000		MOV	#13, RMCS1(RO)		;RELEASE THE DRIVE
4415	023004	104026				ERROR	26		;TYPE ERROR MESSAGE 26
4416	023006	013737	001170	001126	76\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4417	023014	013737	001224	001240		MOV	PORTA, PTNBR		;CHANGE PORT NUMBER
4418	023022	042737	100000	001126		BIC	#ATA,\$BDDAT		;DON'T CHECK THE ATTN BIT
4419	023030	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;ALL BITS OK ?
4420	023036	001401				BEG	77\$;BR IF OK FROM PORT A.
4421	023040	104007				ERROR	7		;REPORT ERROR

```

4422 023042 013737 001172 001126 78S: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4423 023050 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4424 023056 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4425 023064 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4426 023072 001401 BEQ 78S ;BR IF OK
4427 023074 104007 ERROR 7 ;REPORT ERROR
4428 023076 000240 78S: NOP
4429 023100 000004 1S: SCOPE ;LOOP ?

```

```

4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477

```

```

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

```

```

T16:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2S ;BR IF NOT
1S ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2S: MOVB #16,$TSTNM ;TEST NUMBER
MOV #TEST16,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST16,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST16: MOV #STACK,SP ;LOAD THE STACK POINTER

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

```

```

*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RMCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RMCS1(RO) ;READ RMCS1
MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPR ;'OPPOSITE' PORT ADDRESS
MOV RMDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A

```

H07

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 85
T16 SEIZE 'A' BY RMCS1 TEST

SEQ 0085

```

4478 023304 010037 001122      MOV      RO,$BDAOR      ;RHL1 BASE ADDRESS
4479 023310 062737 000012 001122  ADD      #RMDS1,$BDAOR  ;GENERATE BAD REGISTER ADDRESS
4480 023316 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
4481 023322 023737 001124 001126  CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
4482 023330 001403      BEQ      64$           ;BR IF IT IS
4483 023332 104004      ERROR   4             ;REPORT THE ERROR
4484 023334 000137 024024      JMP      1$           ;BYPASS REST OF THE SUBTEST
4485 023340
4486 023340 113760 001224 000010 64$:  MOV     PORTA, RMCS2(RO) ;SELECT PORT A
4487 023346 013737 001224 001240  MOV     PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4488 023354 016037 000012 001126  MOV     RMDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4489 023362 042737 020001 001126  BIC     #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
4490 023370 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
4491 023376 013737 001124 001166  MOV     $GDDAT,$STMP1 ;USE GOOD DATA AS A MASK
4492 023404 005137 001166      COM     $STMP1        ;COMPLEMENT THE EXPECTED STATUS
4493 023410 013737 001126 001164  MOV     $BDDAT,$STMP0 ;SAVE THE ACTUAL STATUS
4494 023416 043737 001166 001164  BIC     $STMP1,$STMP0 ;CLEAR UNWANTED BITS
4495 023424 023737 001124 001164  CMP     $GDDAT,$STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4496 023432 001401      BEQ     65$           ;BR IF THEY ARE
4497 023434 104005      ERROR  5             ;REPORT THE ERROR
4498 023436 000240 65$:  NOP
4499
4500 ;:*****
4501
4502 ;RELEASE THE DRIVE FROM PORT A
4503
4504 023440 113760 001224 000010  MOV     PORTA, RMCS2(RO) ;SELECT PORT A
4505 023446 013737 001224 001240  MOV     PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4506 023454 012760 000013 000000  MOV     #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
4507
4508 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4509
4510 023462 005037 001254      CLR     RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
4511 023466 012737 000012 001122  MOV     #RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4512 023474 060037 001122      ADD     RO,$BDAOR     ;ADD THE I/O BASE ADDRESS
4513 023500 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4514 023506 113760 001224 000010  MOV     PORTA, RMCS2(RO) ;SELECT PORT A.
4515 023514 016037 000012 001170  MOV     RMDS1(RO), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4516 023522 042737 024001 001170  BIC     #PIP!WRL!OM,$STMP2 ;CLEAR DONT CARES
4517 023530 013737 001170 001164  MOV     $STMP2,$STMP0 ;COPY IT INTO 'STMP0'
4518 023536 042737 100100 001164  BIC     #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4519 023544 113760 001226 000010  MOV     PORTB, RMCS2(RO) ;SELECT PORT B.
4520 023552 016037 000012 001172  MOV     RMDS1(RO), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4521 023560 042737 024001 001172  BIC     #PIP!WRL!OM,$STMP3 ;CLEAR DONT CARES
4522 023566 013737 001172 001166  MOV     $STMP3,$STMP1 ;COPY IT INTO 'STMP1'
4523 023574 042737 100100 001166  BIC     #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4524 023602 023737 001164 001166  CMP     $STMP0,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4525 023610 001006      BNE    66$           ;BR IF NOT
4526 023612 005737 001164      TST    $STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4527 023616 001045      BNE    68$           ;BR IF NOT
4528 023620 104046      ERROR  46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4529 023622 000137 024022      JMP    70$           ;BYPASS THE REST OF THE CHECKS
4530 023626 013737 001170 001126 66$:  MOV     $STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4531 023634 013737 001226 001240  MOV     PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4532 023642 113760 001226 000010  MOV     PORTB, RMCS2(RO) ;SELECT PORT B.
4533 023650 005737 001164      TST    $STMP0        ;SEE IF STATUS EQ 0 FROM PORT A.

```

```

4534 023654 001414          BEQ      67$          ;BR IF ZERO
4535 023656 013737 001224 001240    MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4536 023664 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4537 023672 113760 001224 000010    MOVVB  PORTA,AMCS2(RO) ;SELECT PORT A.
4538 023700 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
4539 023704 001012          BNE     68$          ;BR IF NOT
4540 023706 012737 177777 001254 67$:   MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
4541 023714 012760 000011 000000    MOV     #11,AMCS1(RO) ;CLEAR THE DRIVE
4542 023722 012760 000013 000000    MOV     #13,AMCS1(RO) ;RELEASE THE DRIVE
4543 023730 104026          ERROR   26          ;TYPE ERROR MESSAGE 26
4544 023732 013737 001170 001126 68$:   MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4545 023740 013737 001224 001240    MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
4546 023746 042737 100000 001126    BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
4547 023754 023737 001124 001126    CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
4548 023762 001401          BEQ     69$          ;BR IF OK FROM PORT A.
4549 023764 104007          ERROR   7          ;REPORT ERROR
4550 023766 013737 001172 001126 69$:   MOV     $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4551 023774 013737 001226 001240    MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
4552 024002 042737 100000 001126    BIC     #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
4553 024010 023737 001124 001126    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4554 024016 001401          BEQ     70$          ;BR IF OK
4555 024020 104007          ERROR   7          ;REPORT ERROR
4556 024022 000240 70$:   NOP
4557 024024 000004 71$:   NOP                ;LOOP ?
4558
4559 ;*****
4560 ;*TEST 17 SEIZE 'B' BY RMCS1 TEST
4561 ;*
4562 ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
4563 ;*IF THE DRIVE IS IN NEUTRAL.
4564 ;* A. READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT
4565 ;* THE DRIVE IS SEIZED.
4566 ;*
4567 ;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
4568 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4569 ;*
4570 ;*****
4571 ;*TEST17:
4572 024026          TST     KYBCTL       ;PERFORMING ONLY SINGLE TESTS ?
4573 024032 001406 001300    BEQ     1$          ;BR IF NOT
4574 024034 100002          BPL     1$          ;BR IF JUST ENTERED TEST
4575 024036 000137 002676    JMP     EXEC         ;RETURN & GET NEXT TEST NUMBER
4576 024042 012737 177777 001300 1$:   MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
4577 024050 112737 000017 001102 2$:   MOVVB  #17,$STSTNM  ;TEST NUMBER
4578 024056 012737 024100 001106    MOV     #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
4579 024064 012737 024100 001110    MOV     #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
4580 024072 012737 000031 001176    MOV     #25,$TIMES   ;DO 25 ITERATIONS
4581 024100 012706 001100  TEST17: MOV     #STACK,SP   ;LOAD THE STACK POINTER
4582
4583 ;CLEAR ATTENTION BITS FOR BOTH PORTS
4584
4585 024104 113760 001224 000010    MOVVB  PORTA,AMCS2(RO) ;SELECT PORT #A
4586 024112 005060 000012 000000    CLR     RMDS1(RO)    ;SEIZE THE DRIVE
4587 024116 012760 000011 000000    MOV     #11,AMCS1(RO) ;ISSUE DRIVE CLEAR
4588 024124 012760 000013 000000    MOV     #13,AMCS1(RO) ;RELEASE THE DRIVE
4589 024132 113760 001226 000010    MOVVB  PORTB,AMCS2(RO) ;SELECT PORT #B

```



```

4590 024140 005060 000012          CLR      RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
4591 024144 012760 000011 000000  MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
4592 024152 012760 000013 000000  MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
4593
4594 ;:*****
4595
4596 ;SEIZE THE DRIVE THROUGH PORT B
4597
4598 024160 113760 001226 000010  MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
4599 024166 013737 001226 001242  MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
4600 024174 005760 000000          TST      RMCS1(RO)      ;READ RMCS1
4601 024200 113760 001224 000010  MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
4602 024206 013737 001224 001240  MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4603 024214 013737 001224 001244  MOV      PORTA,OPRT ;'OPPOSITE' PORT ADDRESS
4604 024222 016037 000012 001126  MOV      RMDS1(RO),%BDDAT ;SEE IF DRIVE SEIZED BY PORT B
4605 024230 010037 001122          MOV      RO,%BDADR ;R#11 BASE ADDRESS
4606 024234 062737 000012 001122  ADD      #RMDS1,%BDADR ;GENERATE BAD REGISTER ADDRESS
4607 024242 005037 001124          CLR      %GDDAT ;REGISTER SHOULD BE ZERO
4608 024246 023737 001124 001126  CMP      %GDDAT,%BDDAT ;IS THE REGISTER ZERO
4609 024254 001403          BEQ      64$ ;BR IF IT IS
4610 024256 104004          ERROR   4 ;REPORT THE ERROR
4611 024260 000137 024750          JMP      1$ ;BYPASS REST OF THE SUBTEST
4612 024264
4613 024264 113760 001226 000010 64$: MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
4614 024272 013737 001226 001240  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4615 024300 016037 000012 001126  MOV      RMDS1(RO),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4616 024306 042737 020001 001126  BIC      #OM!PIP,%BDDAT ;CLEAR DONT CARE BITS
4617 024314 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,%GDDAT ;EXPECTED STATUS
4618 024322 013737 001124 001166  MOV      %GDDAT,%STMP1 ;USE GOOD DATA AS A MASK
4619 024330 005137 001166          COM      %STMP1 ;COMPLEMENT THE EXPECTED STATUS
4620 024334 013737 001126 001164  MOV      %BDDAT,%STMP0 ;SAVE THE ACTUAL STATUS
4621 024342 043737 001166 001164  BIC      %STMP1,%STMP0 ;CLEAR UNWANTED BITS
4622 024350 023737 001124 001164  CMP      %GDDAT,%STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
4623 024356 001401          BEQ      65$ ;BR IF THEY ARE
4624 024360 104005          ERROR   5 ;REPORT THE ERROR
4625 024362 000240          NOP
4626
4627 ;:*****
4628
4629 ;RELEASE THE DRIVE FROM PORT B
4630
4631 024364 113760 001226 000010  MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
4632 024372 013737 001226 001240  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4633 024400 012760 000013 000000  MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4634
4635 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4636
4637 024406 005037 001254          CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4638 024412 012737 000012 001122  MOV      #RMDS1,%BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4639 024420 060037 001122          ADD      RO,%BDADR ;ADD THE I/O BASE ADDRESS
4640 024424 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,%GDDAT ;COMPARISON CONSTANT
4641 024432 113760 001224 000010  MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
4642 024440 016037 000012 001170  MOV      RMDS1(RO),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
4643 024446 042737 024001 001170  BIC      #PIP!WRL!OM,%STMP2 ;CLEAR DONT CARES
4644 024454 013737 001170 001164  MOV      %STMP2,%STMP0 ;COPY IT INTO 'STMP0'
4645 024462 042737 100100 001164  BIC      #ATA!VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```



```

4646 024470 113760 001226 000010      MOV  PORTB, RMCS2(RO) ; SELECT PORT B.
4647 024476 016037 000012 001172      MOV  RMDS1(RO), $TMP3 ; GET THE DRIVE STATUS REGISTER FROM PORT B.
4648 024504 042737 024001 001172      BIC  #PIP:WRL:OM, $TMP3 ; CLEAR DONT CARES
4649 024512 013737 001172 001166      MOV  $TMP3, $TMP1 ; COPY IT INTO '$TMP1'
4650 024520 042737 100100 001166      BIC  #ATA:VV, $TMP1 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
4651 024526 023737 001164 001166      CMP  $TMP0, $TMP1 ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4652 024534 001006      BNE  66$ ; BR IF NOT
4653 024536 005737 001164      TST  $TMP0 ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
4654 024542 001045      BNE  68$ ; BR IF NOT
4655 024544 104046      ERROR 46 ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4656 024546 000137 024746      JMP  70$ ; BYPASS THE REST OF THE CHECKS
4657 024552 013737 001170 001126 66$: MOV  $TMP2, $BDDAT ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4658 024560 013737 001226 001240      MOV  PORTB, PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4659 024566 113760 001226 000010      MOV  PORTB, RMCS2(RO) ; SELECT PORT B.
4660 024574 005737 001164      TST  $TMP0 ; SEE IF STATUS EQ 0 FROM PORT A.
4661 024600 001414      BEQ  67$ ; BR IF ZERO
4662 024602 013737 001224 001240      MOV  PORTA, PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4663 024610 013737 001172 001126      MOV  $TMP3, $BDDAT ; 'BAD DATA' FOR ERROR TYPE OUT
4664 024616 113760 001224 000010      MOV  PORTA, RMCS2(RO) ; SELECT PORT A.
4665 024624 005737 001166      TST  $TMP1 ; SEE IF STATUS EQ ZERO FROM PORT B.
4666 024630 001012      BNE  68$ ; BR IF NOT
4667 024632 012737 177777 001254 67$: MOV  #-1, RELERR ; SET 'RELEASE ERROR' INDICATOR
4668 024640 012760 000011 000000      MOV  #11, RMCS1(RO) ; CLEAR THE DRIVE
4669 024646 012760 000013 000000      MOV  #13, RMCS1(RO) ; RELEASE THE DRIVE
4670 024654 104026      ERROR 26 ; TYPE ERROR MESSAGE 26
4671 024656 013737 001170 001126 68$: MOV  $TMP2, $BDDAT ; LOOK FOR BIT FAILURES WHEN RMDS1 READ
4672 024664 013737 001224 001240      MOV  PORTA, PTNBR ; CHANGE PORT NUMBER
4673 024672 042737 100000 001126      BIC  #ATA, $BDDAT ; DON'T CHECK THE ATTN BIT
4674 024700 023737 001124 001126      CMP  $GDDAT, $BDDAT ; ALL BITS OK ?
4675 024706 001401      BEQ  69$ ; BR IF OK FROM PORT A.
4676 024710 104007      ERROR 7 ; REPORT ERROR
4677 024712 013737 001172 001126 69$: MOV  $TMP3, $BDDAT ; CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4678 024720 013737 001226 001240      MOV  PORTB, PTNBR ; CHANGE PORT NUMBER
4679 024726 042737 100000 001126      BIC  #ATA, $BDDAT ; DON'T CHECK THE ATTN BIT
4680 024734 023737 001124 001126      CMP  $GDDAT, $BDDAT ; SEE IF READ OK FROM PORT B.
4681 024742 001401      BEQ  70$ ; BR IF OK
4682 024744 104007      ERROR 7 ; REPORT ERROR
4683 024746 000240      NOP ; LOOP ?
4684 024750 000004      IS:  SCOPE

```

```

4685
4686
4687
4688 *****
4689 *TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST
4690 *
4691 *VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
4692 * REQUEST' IF THE DRIVE IS SEIZED.
4693 *
4694 * A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1. VERIFY THAT
4695 * THE DRIVE HAS BEEN SEIZED.
4696 *
4697 * B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT
4698 * SET.
4699 *
4700 * C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
4701 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

```

```

4702
4703 024752
4704 024752 005737 001300
4705 024756 001406
4706 024760 100002
4707 024762 000137 002676
4708 024766 012737 177777 001300 1$:
4709 024774 112737 000020 001102 2$:
4710 025002 012737 025024 001106
4711 025010 012737 025024 001110
4712 025016 012737 000031 001176
4713 025024 012706 001100
4714
4715
4716
4717 025030 113760 001224 000010
4718 025036 005060 000012
4719 025042 012760 000011 000000
4720 025050 012760 000013 000000
4721 025056 113760 001226 000010
4722 025064 005060 000012
4723 025070 012760 000011 000000
4724 025076 012760 000013 000000
4725
4726
4727
4728
4729
4730 025104 113760 001226 000010
4731 025112 013737 001226 001242
4732 025120 005760 000000
4733 025124 113760 001224 000010
4734 025132 013737 001224 001240
4735 025140 013737 001224 001244
4736 025146 016037 000012 001126
4737 025154 010037 001122
4738 025160 062737 000012 001122
4739 025166 015037 001124
4740 025172 023737 001124 001126
4741 025200 001403
4742 025202 104004
4743 025204 000137 026016
4744 025210
4745 025210 113760 001226 000010
4746 025216 013737 001226 001240
4747 025224 016037 000012 001126
4748 025232 042737 020001 001126
4749 025240 012737 011700 001124
4750 025246 013737 001124 001166
4751 025254 005137 001166
4752 025260 013737 001126 001164
4753 025266 043737 001166 001164
4754 025274 023737 001124 001164
4755 025302 001401
4756 025304 104005
4757 025306 000240

```

```

*****
↑ST20:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #20,$STNM ;TEST NUMBER
MOV #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST20: MOV #STACK,SP ;LOAD THE STACK POINTER

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

*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RMCS1(RO) ;READ RMCS1
MOVB PORTA,RMCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPP-SITE' PORT ADDRESS
MOV RMDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$BDADR ;R#11 BASE ADDRESS
ADD #RMDS1,$BDADR ;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 15 ;BYPASS REST OF THE SUBTEST

64$:
MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
BIC #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ^
BEQ 65$ ;BR IF THEY ARE
ERROR 5 ;REPORT THE ERROR

65$:
NOP

```

M07

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 90
T20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST

SEQ 0090

```

4758 025310 113760 001224 000010      MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
4759 025316 013737 001224 001240      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4760
4761
4762
4763
4764 025324 005037 001250      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4765 025330 016037 000000 001126      MOV   RMCS1(RO),SBDDAT ;GET CONTENTS OF RMCS1
4766 025336 012737 000000 001122      MOV   #RMCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4767 025344 060037 001122      ADD   RO,SBADR ;ADD RHL1 BASE ADDRESS
4768 025350 005037 001124      CLR   $GDDAT ;WHAT REGISTER SHOULD BE
4769 025354 013737 001126 001164      MOV   SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4770 025362 042737 173700 001164      BIC   #1C4077,$TMP0 ;SAVE SPECIFIED BITS
4771 025370 023737 001124 001164      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
4772 025376 001414 66$ ;BR IF OK
4773 025400 013737 001126 001174      MOV   SBDDAT,$TMP4 ;COPY 'BAD DATA'
4774 025406 042737 004077 001174      BIC   #4077,$TMP4 ;CLEAR THE MASKED BITS
4775 025414 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4776 025422 104010 001174 001124      ERROR 10 ;REPORT THE ERROR
4777 025424 005137 001250      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4778 025430 000240 66$: NOP
4779
4780
4781
4782
4783
4784
4785
4786 025432 113760 001226 000010      MOVE  PORTB,RMCS2(RO) ;SELECT PORT B
4787 025440 013737 001226 001240      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4788 025446 012760 000013 000000      MOV   #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813

```

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

;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

;CLEAR THE 'RELEASE ERROR' INDICATOR
;FORM THE ADDRESS OF RMOS1 FOR TYPEOUT
;ADD THE I/O BASE ADDRESS
;COMPARISON CONSTANT
;SELECT PORT A.
;GET THE DRIVE STATUS REGISTER FROM PORT A.
;CLEAR DONT CARES
;COPY IT INTO '$TMP0'
;CLEAR PORT DEPENDENT BITS FROM THE COPY
;SELECT PORT B.
;GET THE DRIVE STATUS REGISTER FROM PORT B.
;CLEAR DONT CARES
;COPY IT INTO '$TMP1'
;CLEAR PORT DEPENDENT BITS FROM THE COPY
;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
;BR IF NOT
;REGISTERS ARE THE SAME: ARE THEY ZERO ?
;BR IF NOT
;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
;BYPASS THE REST OF THE CHECKS
;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

4814 025634 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ; SELECT PORT B.
4815 025642 005737 001164              TST    $TMP0           ; SEE IF STATUS EQ 0 FROM PORT A.
4816 025646 001414              BEQ    69$             ; BR IF ZERO
4817 025650 013737 001224 001240      MOV    PORTA, PTNBR    ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4818 025656 013737 001172 001126      MOV    $TMP3, $BDDAT   ; 'BAD DATA' FOR ERROR TYPE OUT
4819 025664 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ; SELECT PORT A.
4820 025672 005737 001166              TST    $TMP1           ; SEE IF STATUS EQ ZERO FROM PORT B.
4821 025676 001012              BNE    70$            ; BR IF NOT
4822 025700 012737 177777 001254 69$:      MOV    #-1, RELERR    ; SET 'RELEASE ERROR' INDICATOR
4823 025706 012760 000011 000000      MOV    #11, RMCS1(R0) ; CLEAR THE DRIVE
4824 025714 012760 000013 000000      MOV    #13, RMCS1(R0) ; RELEASE THE DRIVE
4825 025722 104026              ERROR  26             ; TYPE ERROR MESSAGE 26
4826 025724 013737 001170 001126 70$:      MOV    $TMP2, $BDDAT   ; LOOK FOR BIT FAILURES WHEN RMDS1 READ
4827 025732 013737 001224 001240      MOV    PORTA, PTNBR    ; CHANGE PORT NUMBER
4828 025740 042737 100000 001126      BIC    #ATA, $BDDAT    ; DON'T CHECK THE ATTN BIT
4829 025746 023737 001124 001126      CMP    $GDDAT, $BDDAT  ; ALL BITS OK ?
4830 025754 001401              BEQ    71$            ; BR IF OK FROM PORT A.
4831 025756 104007              ERROR  7              ; REPORT ERROR
4832 025760 013737 001172 001126 71$:      MOV    $TMP3, $BDDAT   ; CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4833 025766 013737 001226 001240      MOV    PORTB, PTNBR    ; CHANGE PORT NUMBER
4834 025774 042737 100000 001126      BIC    #ATA, $BDDAT    ; DON'T CHECK THE ATTN BIT
4835 026002 023737 001124 001126      CMP    $GDDAT, $BDDAT  ; SEE IF READ OK FROM PORT B.
4836 026010 001401              BEQ    72$            ; BR IF OK
4837 026012 104007              ERROR  7              ; REPORT ERROR
4838 026014 000240 72$:      NOP
4839 026016 000004 1$:      SCOPE ; LOOP ?

```

```

4840
4841
4842 :*****
4843 *TEST 21 PORT 'B' INHIBIT SEIZE BY RMCS1 TEST
4844 *
4845 *VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
4846 * REQUEST' IF THE DRIVE IS SEIZED.
4847 *
4848 * A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1. VERIFY THAT
4849 * THE DRIVE HAS BEEN SEIZED.
4850 *
4851 * B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT
4852 * SET.
4853 *
4854 * C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
4855 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4856 *
4857 :*****

```

```

4858 026020 005737 001300      †ST21: TST    KYBCTL          ; PERFORMING ONLY SINGLE TESTS ?
4859 026024 001406              BEQ    25             ; BR IF NOT
4860 026026 100002              BPL    1$            ; BR IF JUST ENTERED TEST
4861 026030 000137 002676              JMP    EXEC           ; RETURN & GET NEXT TEST NUMBER
4862 026034 012737 177777 001300 1$:      MOV    #-1, KYBCTL    ; SET SINGLE TEST INDICATOR
4863 026042 112737 000021 001102 2$:      MOVB   #21, $TSTNM    ; TEST NUMBER
4864 026050 012737 026072 001106      MOV    #TEST21, $LPADR ; LOAD LOOP ON TEST ADDRESS
4865 026056 012737 026072 001110      MOV    #TEST21, $LPERR ; LOAD LOOP ON ERROR ADDRESS
4866 026064 012737 000031 001176      MOV    #25, $TIMES    ; DO 25 ITERATIONS
4867 026072 012706 001100      TEST21: MOV    #STACK, SP    ; LOAD THE STACK POINTER
4868
4869 ;CLEAR ATTENTION BITS FOR BOTH PORTS

```

```

4870
4871 026076 113760 001224 000010      MOV      PORTA, RMCS2(RO) ; SELECT PORT #A
4872 026104 005060 000012 000000      CLR      RMDS1(RO)      ; SEIZE THE DRIVE
4873 026110 012760 000011 000000      MOV      #11, RMCS1(RO) ; ISSUE DRIVE CLEAR
4874 026116 012760 000013 000000      MOV      #13, RMCS1(RO) ; RELEASE THE DRIVE
4875 026124 113760 001226 000010      MOV      PORTB, RMCS2(RO) ; SELECT PORT #B
4876 026132 005060 000012 000000      CLR      RMDS1(RO)      ; SEIZE THE DRIVE THROUGH PORT 'B'
4877 026136 012760 000011 000000      MOV      #11, RMCS1(RO) ; ISSUE DRIVE CLEAR
4878 026144 012760 000013 000000      MOV      #13, RMCS1(RO) ; RELEASE THE DRIVE
4879
4880 ;:*****
4881
4882 ; SEIZE THE DRIVE THROUGH PORT A
4883
4884 026152 113760 001224 000010      MOV      PORTA, RMCS2(RO) ; SELECT PORT A
4885 026160 013737 001224 001242      MOV      PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
4886 026166 005760 000000 000000      TST      RMCS1(RO)      ; READ RMCS1
4887 026172 113760 001226 000010      MOV      PORTB, RMCS2(RO) ; SELECT PORT B
4888 026200 013737 001226 001240      MOV      PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4889 026206 013737 001226 001244      MOV      PORTB, OPPRT ; 'OPPOSITE' PORT ADDRESS
4890 026214 016037 000012 001126      MOV      RMDS1(RO), $BDDAT ; SEE IF DRIVE SEIZED BY PORT A
4891 026222 010037 001122 001122      MOV      RO, $BDAOR ; RHI1 BASE ADDRESS
4892 026226 062737 000012 001122      ADD      #RMDS1, $BDAOR ; GENERATE BAD REGISTER ADDRESS
4893 026234 005037 001124 001126      CLR      $GDDAT ; REGISTER SHOULD BE ZERO
4894 026240 023737 001124 001126      CMP      $GDDAT, $BDDAT ; IS THE REGISTER ZERO
4895 026246 001403 001124 001126      BEQ      64$ ; BR IF IT IS
4896 026250 104004 001124 001126      ERROR   4 ; REPORT THE ERROR
4897 026252 000137 027064 001126      JMP      1$ ; BYPASS REST OF THE SUBTEST
4898
4899 026256 113760 001224 000010      MOV      PORTA, RMCS2(RO) ; SELECT PORT A
4900 026264 013737 001224 001240      MOV      PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4901 026272 016037 000012 001126      MOV      RMDS1(RO), $BDDAT ; SEE IF SEIZING PORT SEES CORRECT STATUS
4902 026300 042737 020001 001126      BIC      #OM:PIP, $BDDAT ; CLEAR DONT CARE BITS
4903 026306 012737 011700 001124      MOV      #MOL:PGM:DPR:DRY:VV, $GDDAT ; EXPECTED STATUS
4904 026314 013737 001124 001166      MOV      $GDDAT, $TMP1 ; USE GOOD DATA AS A MASK
4905 026322 005137 001166 001166      COM      $TMP1 ; COMPLEMENT THE EXPECTED STATUS
4906 026326 013737 001126 001164      MOV      $BDDAT, $TMP0 ; SAVE THE ACTUAL STATUS
4907 026334 043737 001166 001164      BIC      $TMP1, $TMP0 ; CLEAR UNWANTED BITS
4908 026342 023737 001124 001164      CMP      $GDDAT, $TMP0 ; ARE THE EXPECTED STATUS BITS SET ?
4909 026350 001401 001124 001164      BEQ      65$ ; BR IF THEY ARE
4910 026352 104005 001124 001164      ERROR   5 ; REPORT THE ERROR
4911
4912 026354 000240 001226 000010      NOP
4913 026356 113760 001226 000010      MOV      PORTB, RMCS2(RO) ; SELECT PORT B
4914 026364 013737 001226 001240      MOV      PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4915
4916 ;:*****
4917 ; READ RMCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
4918
4918 026372 005037 001250 001126      CLR      CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
4919 026376 016037 000000 001126      MOV      RMCS1(RO), $BDDAT ; GET CONTENTS OF RMCS1
4920 026404 012737 000000 001122      MOV      #RMCS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
4921 026412 060037 001122 001122      ADD      RO, $BDAOR ; ADD RHI1 BASE ADDRESS
4922 026416 005037 001124 001164      CLR      $GDDAT ; WHAT REGISTER SHOULD BE
4923 026422 013737 001126 001164      MOV      $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
4924 026430 042737 173700 001164      BIC      #1C40??, $TMP0 ; SAVE SPECIFIED BITS
4925 026436 023737 001124 001164      CMP      $GDDAT, $TMP0 ; COMPARE THE BITS

```

4926	026444	001414				BEQ	66\$;BR IF OK
4927	026446	013737	001126	001174		MOV	\$BDDAT,\$TMP4		;COPY 'BAD DATA'
4928	026454	042737	004077	001174		BIC	#4077,\$TMP4		;CLEAR THE MASKED BITS
4929	026462	053737	001174	001124		BIS	\$TMP4,\$GDDAT		; 'OR' WITH GOOD DATA FOR TYPEOUT
4930	026470	104010				ERROR	10		;REPORT THE ERROR
4931	026472	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
4932	026476	000240				66\$:	NOP		
4933									
4934									
4935									;*****
4936									;DRIVE SHOULD RETURN TO NEUTRAL
4937									
4938									;RELEASE THE DRIVE FROM PORT A
4939									
4940	026500	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A
4941	026506	013737	001224	001240		MOV	PORTA,PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4942	026514	012760	000013	000000		MOV	#13,RMCS1(RO)		;ISSUE RELEASE THROUGH PORT A
4943									
4944									;VERIFY THAT THE DRIVE IS IN NEUTRAL
4945									
4946	026522	005037	001254			CLR	RELERR		;CLEAR THE 'RELEASE ERROR' INDICATOR
4947	026526	012737	000012	001122		MOV	#RMDS1,\$BDDAR		;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
4948	026534	060037	001122			ADD	RO,\$BDDAR		;ADD THE I/O BASE ADDRESS
4949	026540	012737	011700	001124		MOV	#MOL,PCN:OPR:DRY:VV,\$GDDAT		;COMPARISON CONSTANT
4950	026546	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A.
4951	026554	016037	000012	001170		MOV	RMDS1(RO),\$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.
4952	026562	042737	024001	001170		BIC	#PIP:WAL:OM,\$TMP2		;CLEAR DONT CARES
4953	026570	013737	001170	001164		MOV	\$TMP2,\$TMP0		;COPY IT INTO '\$TMP0'
4954	026576	042737	100100	001164		BIC	#ATA:VV,\$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4955	026604	113760	001226	000010		MOVB	PORTB,RMCS2(RO)		;SELECT PORT B.
4956	026612	016037	000012	001172		MOV	RMDS1(RO),\$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.
4957	026620	042737	024001	001172		BIC	#PIP:WAL:OM,\$TMP3		;CLEAR DONT CARES
4958	026626	013737	001172	001166		MOV	\$TMP3,\$TMP1		;COPY IT INTO '\$TMP1'
4959	026634	042737	100100	001166		BIC	#ATA:VV,\$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY
4960	026642	023737	001164	001166		CMP	\$TMP0,\$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4961	026650	001006				BNE	68\$;BR IF NOT
4962	026652	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4963	026656	001045				BNE	70\$;BR IF NOT
4964	026660	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4965	026662	000137	027062			JMP	72\$;BYPASS THE REST OF THE CHECKS
4966	026666	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4967	026674	013737	001226	001240		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4968	026702	113760	001226	000010		MOVB	PORTB,RMCS2(RO)		;SELECT PORT B.
4969	026710	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
4970	026714	001414				BEQ	69\$;BR IF ZERO
4971	026716	013737	001224	001240		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4972	026724	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
4973	026732	113760	001224	000010		MOVB	PORTA,RMCS2(RO)		;SELECT PORT A.
4974	026740	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
4975	026744	001012				BNE	70\$;BR IF NOT
4976	026746	012737	177777	001254	69\$:	MOV	#-1,RELERR		;SET 'RELEASE ERROR' INDICATOR
4977	026754	012760	000011	000000		MOV	#11,RMCS1(RO)		;CLEAR THE DRIVE
4978	026762	012760	000013	000000		MOV	#13,RMCS1(RO)		;RELEASE THE DRIVE
4979	026770	104026				ERROR	26		;TYPE ERROR MESSAGE 26
4980	026772	013737	001170	001126	70\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RMDS1 READ
4981	027000	013737	001224	001240		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER

```

4982 027006 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4983 027014 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
4984 027022 001401 BEQ 71$ ;BR IF OK FROM PORT A.
4985 027024 104007 ERROR 7 ;REPORT ERROR
4986 027026 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
4987 027034 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4988 027042 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
4989 027050 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
4990 027056 001401 BEQ 72$ ;BR IF OK
4991 027060 104007 ERROR 7 ;REPORT ERROR
4992 027062 000240 72$: NOP
4993 027064 000004 1$: SCOPE ;LOOP ?

```

```

4994
4995
4996
4997 *****
4998 *TEST 22 SEIZE BY RMAS TEST
4999
5000 *TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
5001 * (RMAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER
5002 * PORT.
5003
5004 * A. WRITE THE APPROPRIATE DRIVE BIT INTO RMAS; VERIFY THAT THE DRIVE
5005 * IS SEIZED.
5006
5007 * B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
5008 * DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE
5009 * OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
5010

```

```

5011 027066 *****
5012 027066 005737 001300 †ST22: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5013 027072 001406 BEQ 2$ ;BR IF NOT
5014 027074 100002 BPL 1$ ;BR IF JUST ENTERED TEST
5015 027076 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5016 027102 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
5017 027110 112737 000022 001102 2$: MOVB #22,$STNM ;TEST NUMBER
5018 027116 012737 027140 001106 MOV #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS
5019 027124 012737 027140 001110 MOV #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5020 027132 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
5021 027140 012706 001100 TEST22: MOV #STACK,SP ;LOAD THE STACK POINTER

```

```

5022
5023 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5024
5025 027144 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
5026 027152 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
5027 027156 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5028 027164 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5029 027172 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
5030 027200 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
5031 027204 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5032 027212 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE

```

```

5033
5034 *****
5035 ;SELECT DRIVE OTHER THAN THAT BEING TESTED
5036
5037 027220 113760 001230 000010 MOVB PORTC,RMCS2(RO) ;SELECT DRIVE NOT BEING TESTED

```



```

5038 027226 013737 001224 001242      MOV      PORTA,SEIZPT      ;'SEIZED' PORT ADDRESS
5039
5040      ;:*****
5041      ;WRITE THE DRIVE'S ATTENTION BIT
5042
5043 027234 013760 001236 000016      MOV      ASR1,RMAS(RO)    ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
5044 027242 113760 001224 000010      MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A
5045 027250 013737 001224 001240      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5046
5047      ;:*****
5048      ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
5049
5050 027256 005760 000012      TST      RMDS1(RO)       ;SEE THE REGISTER THROUGH PORT A ?
5051 027262 001014      BNE      1$             ;BR IF YES
5052 027264 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
5053 027272 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5054 027300 005760 000012      TST      RMDS1(RO)       ;SEE REGISTER THROUGH PORT B ?
5055 027304 001021      BNE      2$             ;BR IF YES
5056 027306 104042      ERROR   42             ;DRIVE NOT IN NEUTRAL OR SEIZED
5057 027310 000137 031140      JMP      4$             ;BYPASS REST OF TEST
5058 027314
5059 027314 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
5060 027322 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5061 027330 005760 000012      TST      RMDS1(RO)       ;REGISTER SHOULD BE ZERO THROUGH PORT B
5062 027334 001002      BNE      .+6           ;BR IF STATUS REG IS NOT ZERO
5063 027336 000137 030240      JMP      3$             ;STATUS REG IS ZERO
5064 027342 104043      ERROR   43             ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
5065 027344 000137 031140      JMP      4$             ;BYPASS REST OF TEST
5066
5067      ;:*****
5068      ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
5069
5070      2$:
5071 027350 005037 001250      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5072 027354 016037 000012 001126      MOV      RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
5073 027362 012737 000012 001122      MOV      #RMDS1,SBDAOR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5074 027370 060037 001122      ADD      RO,SBDAOR       ;ADD RHI1 BASE ADDRESS
5075 027374 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
5076 027402 013737 001126 001164      MOV      SBDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5077 027410 042737 106077 001164      BIC      #1C71700,$TMP0  ;SAVE SPECIFIED BITS
5078 027416 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
5079 027424 001414      BEQ      64$           ;BR IF OK
5080 027426 013737 001126 001174      MOV      SBDDAT,$TMP4    ;COPY 'BAD DATA'
5081 027434 042737 071700 001174      BIC      #71700,$TMP4    ;CLEAR THE MASKED BITS
5082 027442 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5083 027450 104010      ERROR   10             ;REPORT THE ERROR
5084 027452 005137 001250      COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5085 027456 000240
5086 027460 013737 001226 001242      NOP
5087 027466 013737 001224 001244      MOV      PORTB,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
5088      MOV      PORTA,OPPRT    ;SAME AS ABOVE
5089      ;RELEASE THE DRIVE FROM PORT B
5090
5091 027474 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
5092 027502 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5093 027510 012760 000013 000000      MOV      #13,RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT B

```



```

5094
5095 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5096
5097 027516 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5098 027522 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV $GDDAT ;COMPARISON CONSTANT
5099 027530 012737 000012 001122 MOV #RMDS1,$BDADR ;REGISTER ADDRESS INCREMENT
5100 027536 060037 001122 ADD RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
5101 027542 113760 001224 000010 MOV# PORTA, RMCS2(RO) ;SELECT PORT A
5102 027550 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5103 027556 016037 000012 001164 MOV RMDS1(RO), $TMP0 ;READ STATUS REGISTER FROM PORT A
5104 027564 113760 001226 000010 MOV# PORTB, RMCS2(RO) ;SELECT PORT B
5105 027572 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5106 027600 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;DRIVE STATUS FROM PORT B
5107 027606 001404 BEQ 66$ ;BR IF STATUS FROM PORT B ZERO
5108 027610 005737 001164 TST $TMP0 ;IS STATUS FROM PORT A ZERO ?
5109 027614 001401 BEQ 66$ ;BR IF ZERO
5110 027616 104044 ERROR 44 ;REPORT DRIVE NOT SEIZED BY PORT A
5111 027620 013737 001164 001126 66$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
5112 027626 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5113 027634 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
5114 027642 001401 BEQ 67$ ;BR IF OK
5115 027644 104027 ERROR 27 ;REPORT REGISTER ERROR
5116 027646 000240 67$: NOP
5117
5118 ;RELEASE THE DRIVE FROM PORT A
5119
5120 027650 113760 001224 000010 MOV# PORTA, RMCS2(RO) ;SELECT PORT A
5121 027656 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5122 027664 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5123
5124 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5125
5126 027672 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5127 027676 012737 000012 001122 MOV #RMDS1,$BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5128 027704 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
5129 027710 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV $GDDAT ;COMPARISON CONSTANT
5130 027716 113760 001224 000010 MOV# PORTA, RMCS2(RO) ;SELECT PORT A.
5131 027724 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5132 027732 042737 024001 001170 BIC #PIP!WAL!OM, $TMP2 ;CLEAR DONT CARES
5133 027740 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5134 027746 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5135 027754 113760 001226 000010 MOV# PORTB, RMCS2(RO) ;SELECT PORT B.
5136 027762 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5137 027770 042737 024001 001172 BIC #PIP!WAL!OM, $TMP3 ;CLEAR DONT CARES
5138 027776 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5139 030004 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5140 030012 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5141 030020 001006 BNE 68$ ;BR IF NOT
5142 030022 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5143 030026 001045 BNE 70$ ;BR IF NOT
5144 030030 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5145 030032 000137 030232 001126 68$: JMP 72$ ;BYPASS THE REST OF THE CHECKS
5146 030036 013737 001170 001240 MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5147 030044 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5148 030052 113760 001226 000010 MOV# PORTB, RMCS2(RO) ;SELECT PORT B.
5149 030060 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.

```

```

5150 030064 001414 BEQ 69$ ;BR IF ZERO
5151 030066 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5152 030074 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5153 030102 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
5154 030110 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5155 030114 001012 BNE 70$ ;BR IF NOT
5156 030116 012737 177777 001254 69$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
5157 030124 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
5158 030132 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5159 030140 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5160 030142 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5161 030150 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5162 030156 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5163 030164 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK?
5164 030172 001401 BEQ 71$ ;BR IF OK FROM PORT A.
5165 030174 104007 ERROR 7 ;REPORT ERROR
5166 030176 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5167 030204 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5168 030212 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5169 030220 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5170 030226 001401 BEQ 72$ ;BR IF OK
5171 030230 104007 ERROR 7 ;REPORT ERROR
5172 030232 000240 72$: NOP
5173 030234 000137 031140 JMP 4$
5174
5175 ;*****
5176 ;THE DRIVE IS SEIZED BY PORT A. VERIFY THAT PORT B HAS PORT REQUEST SET
5177
5178 030240 35:
5179 030240 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5180 030246 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5181 030254 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5182 030260 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
5183 030266 012737 000012 001122 MOV #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5184 030274 060037 001122 ADD RO,$BDDADR ;ADD RHI1 BASE ADDRESS
5185 030300 012737 011700 001124 MOV #M0L!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
5186 030306 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5187 030314 042737 106077 001164 BIC #1C71700,$TMP0 ;SAVE SPECIFIED BITS
5188 030322 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5189 030330 001414 BEQ 73$ ;BR IF OK
5190 030332 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
5191 030340 042737 071700 001174 BIC #71700,$TMP4 ;CLEAR THE MASKED BITS
5192 030346 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5193 030354 104010 ERROR 10 ;REPORT THE ERROR
5194 030356 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5195 030362 000240 73$: NOP
5196 030364 013737 001224 001242 MOV PORTA,SEIZPT ;ADDRESS FOR ERROR MESSAGE
5197 030372 013737 001226 001244 MOV PORTB,OPPRT ;SAME AS ABOVE
5198
5199 ;RELEASE THE DRIVE FROM PORT A
5200
5201 030400 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5202 030406 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5203 030414 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5204
5205 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A

```

```

5206
5207 030422 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5208 030426 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5209 030434 012737 000012 001122 MOV #RMS1,$BDAADR ;REGISTER ADDRESS INCREMENT
5210 030442 060037 001122 ADD RO,$BDAADR ;REGISTER BASE ADDRESS FOR TYPEOUT
5211 030446 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B
5212 030454 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5213 030462 016037 000012 001164 MOV RMS1(RO), $TMP0 ;READ STATUS REGISTER FROM PORT B
5214 030470 113760 001224 000010 MOVVB PORTA, RMCS2(RO) ;SELECT PORT A
5215 030476 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5216 030504 016037 000012 001126 MOV RMS1(RO), $BDDAT ;DRIVE STATUS FROM PORT A
5217 030512 001404 BEQ 75$ ;BR IF STATUS FROM PORT A ZERO
5218 030514 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO ?
5219 030520 001401 BEQ 75$ ;BR IF ZERO
5220 030522 104044 ERROR 44 ;REPORT DRIVE NOT SEIZED BY PORT B
5221 030524 013737 001164 001126 75$: MOV $TMP0, $BDDAT ;CHECK STATUS FROM PORT B
5222 030532 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5223 030540 023737 001124 001126 CMP $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
5224 030546 001401 BEQ 76$ ;BR IF OK
5225 030550 104027 ERROR 27 ;REPORT REGISTER ERROR
5226 030552 000240 76$: NOP
5227
5228 ;RELEASE THE DRIVE FROM PORT B
5229
5230 030554 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B
5231 030562 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5232 030570 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5233
5234 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5235
5236 030576 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5237 030602 012737 000012 001122 MOV #RMS1,$BDAADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
5238 030610 060037 001122 ADD RO,$BDAADR ;ADD THE I/O BASE ADDRESS
5239 030614 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5240 030622 113760 001224 000010 MOVVB PORTA, RMCS2(RO) ;SELECT PORT A
5241 030630 016037 000012 001170 MOV RMS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5242 030636 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
5243 030644 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
5244 030652 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5245 030660 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B
5246 030666 016037 000012 001172 MOV RMS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5247 030674 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
5248 030702 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
5249 030710 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5250 030716 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5251 030724 001006 BNE 77$ ;BR IF NOT
5252 030726 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5253 030732 001045 BNE 79$ ;BR IF NOT
5254 030734 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5255 030736 000137 031136 JMP 81$ ;BYPASS THE REST OF THE CHECKS
5256 030742 013737 001170 001126 77$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5257 030750 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5258 030756 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B
5259 030764 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5260 030770 001414 BEQ 78$ ;BR IF ZERO
5261 030772 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

5262 031000 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5263 031006 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
5264 031014 005737 001166 TST $TMP1 ;SEE IF STATUS EG ZERO FROM PORT B.
5265 031020 001012 BNE 79$ ;BR IF NOT
5266 031022 012737 177777 001254 78$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5267 031030 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
5268 031036 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5269 031044 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5270 031046 013737 001170 001126 79$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5271 031054 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5272 031062 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5273 031070 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
5274 031076 001401 BEQ 80$ ;BR IF OK FROM PORT A.
5275 031100 104007 ERROR 7 ;REPORT ERROR
5276 031102 013737 001172 001126 80$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5277 031110 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5278 031116 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5279 031124 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5280 031132 001401 BEQ 81$ ;BR IF OK
5281 031134 104007 ERROR 7 ;REPORT ERROR
5282 031136 000240 81$: NOP
5283 031140 000004 4$: SCOPE ;LOOP ?
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317

```

```

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

```

```

5298 031142
5299 031142 005737 001300 †ST23:
5300 031146 001406 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
5301 031150 100002 BEQ 2$ ;BR IF NOT
5302 031152 000137 002676 BPL 1$ ;BR IF JUST ENTERED TEST
5303 031156 012737 177777 001300 1$: JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5304 031164 112737 000023 001102 2$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
5305 031172 012737 031214 001106 MOV #23,$STNM ;TEST NUMBER
5306 031200 012737 031214 001110 MOV #TEST23,$LPADR ;LOAD LOOP ON TEST ADDRESS
5307 031206 012737 000031 001176 MOV #TEST23,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5308 031214 012706 001100 TEST23: MOV #25,$TIMES ;DO 25. ITERATIONS
5309 ;LOAD THE STACK POINTER
5310 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5311
5312 031220 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT #A
5313 031226 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
5314 031232 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5315 031240 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5316 031246 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT #B
5317 031254 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'

```

JOB

CZRMGB0 RMO3/2 DU FOR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 100
T23 INHIBIT SEIZE BY RMAS TEST

SEQ 0100

```

5318 031260 012760 000011 000000      MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
5319 031266 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5320 031274 113760 001230 000010      MOVB    PORTC,RMCS2(RO) ;SELECT DRIVE NOT BEING TESTED
5321
5322      ;*****
5323      ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST
5324
5325 031302 013737 001236 001164      MOV      ASR1,$TMP0 ;STORE ATTN BIT FOR PORT A
5326 031310 005137 001164      COM      $TMP0 ;COMPLEMENT IT
5327 031314 013760 001164 000016      MOV      $TMP0,RMAS(RO) ;WRITE THE ATTN REGISTER
5328
5329      ;*****
5330      ;VERIFY THAT DRIVE REMAINED IN NEUTRAL
5331
5332
5333      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5334
5335 031322 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5336 031326 012737 000012 001122      MOV      #RMDS1,$BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5337 031334 060037 001122      ADD     RO,$BDADR ;ADD THE I/O BASE ADDRESS
5338 031340 012737 011700 001124      MOV      #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
5339 031346 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A.
5340 031354 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5341 031362 042737 024001 001170      BIC     #PIP:WRL:OM,$TMP2 ;CLEAR DONT CARES
5342 031370 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5343 031376 042737 100100 001164      BIC     #ATA:VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5344 031404 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B.
5345 031412 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5346 031420 042737 024001 001172      BIC     #PIP:WRL:OM,$TMP3 ;CLEAR DONT CARES
5347 031426 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5348 031434 042737 100100 001166      BIC     #ATA:VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5349 031442 023737 001164 001166      CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5350 031450 001006      BNE     64$ ;BR IF NOT
5351 031452 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5352 031456 001045      BNE     66$ ;BR IF NOT
5353 031460 104046      ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5354 031462 000137 031662      JMP     68$ ;BYPASS THE REST OF THE CHECKS
5355 031466 013737 001170 001126 64$:      MOV      $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5356 031474 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5357 031502 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B.
5358 031510 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5359 031514 001414      BEQ     65$ ;BR IF ZERO
5360 031516 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5361 031524 013737 001172 001126      MOV      $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
5362 031532 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A.
5363 031540 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5364 031544 001012      BNE     66$ ;BR IF NOT
5365 031546 012737 177777 001254 65$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5366 031554 012760 000011 000000      MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
5367 031562 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
5368 031570 104021      ERROR   21 ;TYPE ERROR MESSAGE 21
5369 031572 013737 001170 001126 66$:      MOV      $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5370 031600 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
5371 031606 042737 100000 001126      BIC     #ATA,$BDAT ;DON'T CHECK THE ATTN BIT
5372 031614 023737 001124 001126      CMP     $GDDAT,$BDAT ;ALL BITS OK ?
5373 031622 001401      BEQ     67$ ;BR IF OK FROM PORT A.

```

K08

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 101
T23 INHIBIT SEIZE BY RMAS TEST

SEQ 0101

5374	031624	104007				ERROR	7	;REPORT ERROR
5375	031626	013737	001172	001126	67\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5376	031634	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
5377	031642	042737	100000	001126		BIC	#ATA,\$BDDAT	;DON'T CHECK THE ATTN BIT
5378	031650	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
5379	031656	001401				BEQ	68\$;BR IF OK
5380	031660	104007				ERROR	7	;REPORT ERROR
5381	031662	000240			68\$:	NOP		
5382	031664	000004				SCOPE		;LOOP ?

```

*****
*TEST 24      SET PORT 'A' REQUEST TEST
*
*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
*DRIVE IS SEIZED BY THE OTHER PORT.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
*
*  B.  WRITE 0'S INTO RMDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
*      SEIZED BY PORT 'B'.
*
*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
*      SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR
*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
*
*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****

```

5404	031666					†ST24:	TST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
5405	031666	005737	001300				BEQ	2\$;BR IF NOT
5406	031672	001406					BPL	1\$;BR IF JUST ENTERED TEST
5407	031674	100002					JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
5408	031676	000137	002676			\$:	MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR
5409	031702	012737	177777	001300	\$:	MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR	
5410	031710	112737	000024	001102	2\$:	MOVB	#24,\$TSTNM	;TEST NUMBER	
5411	031716	012737	031740	001106		MOV	#TEST24,\$LPADR	;LOAD LOOP ON TEST ADDRESS	
5412	031724	012737	031740	001110		MOV	#TEST24,\$LPERR	;LOAD LOOP ON ERROR ADDRESS	
5413	031732	012737	000031	001176		MOV	#25,\$TIMES	;DO 25. ITERATIONS	
5414	031740	012706	001100		TEST24:	MOV	#STACK,SP	;LOAD THE STACK POINTER	

;CLEAR ATTENTION BITS FOR BOTH PORTS

5418	031744	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	;SELECT PORT #A
5419	031752	005060	000012			CLR	RMDS1(RO)	;SEIZE THE DRIVE
5420	031756	012760	000011	000000		MOV	#11,RMCS1(RO)	;ISSUE DRIVE CLEAR
5421	031764	012760	000013	000000		MOV	#13,RMCS1(RO)	;RELEASE THE DRIVE
5422	031772	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	;SELECT PORT #B
5423	032000	005060	000012			CLR	RMDS1(RO)	;SEIZE THE DRIVE THROUGH PORT 'B'
5424	032004	012760	000011	000000		MOV	#11,RMCS1(RO)	;ISSUE DRIVE CLEAR
5425	032012	012760	000013	000000		MOV	#13,RMCS1(RO)	;RELEASE THE DRIVE

;;*****

;SEIZE THE DRIVE THROUGH PORT B

5426
5427
5428
5429

```

5430
5431 032020 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
5432 032026 013737 001226 001242      MOV   PORTB, SEIZPT ; STORE SEIZING PORT'S ADDRESS
5433 032034 005060 000012                CLR   RMDS1(RO) ; WRITE RMDS1
5434 032040 013737 001224 001244      MOV   PORTA, OPPRT ; 'OPPOSITE' PORT ADDRESS
5435 032046 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
5436 032054 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5437
5438 ;*****
5439 ;SET PORT REQUEST
5440
5441 032062 005060 000012                CLR   RMDS1(RO) ; SET PORT REQUEST FOR PORT A
5442
5443 ;*****
5444 ;RELEASE THROUGH PORT B. DRIVE SHOULD SWITCH TO PORT A.
5445
5446
5447 ;RELEASE THE DRIVE FROM PORT B
5448
5449 032066 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
5450 032074 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5451 032102 012760 000013 000000      MOV   #13, RMCS1(RO) ; ISSUE RELEASE THROUGH PORT B
5452
5453 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5454
5455 032110 005037 001254                CLR   RELERR ; CLEAR 'RELEASE ERROR' INDICATOR
5456 032114 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV $GDDAT ; COMPARISON CONSTANT
5457 032122 012737 000012 001122      MOV   #RMDS1, $BDAOR ; REGISTER ADDRESS INCREMENT
5458 032130 060037 001122                ADD   RO, $BDAOR ; REGISTER BASE ADDRESS FOR TYPEOUT
5459 032134 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ; SELECT PORT A
5460 032142 013737 001224 001240      MOV   PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5461 032150 016037 000012 001164      MOV   RMDS1(RO), $TMPO ; READ STATUS REGISTER FROM PORT A
5462 032156 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
5463 032164 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5464 032172 016037 000012 001126      MOV   RMDS1(RO), $BDC T ; DRIVE STATUS FROM PORT B
5465 032200 001404                BEQ   66$ ; BR IF STATUS FROM PORT B ZERO
5466 032202 005737 001164                TST   $TMPO ; IS STATUS FROM PORT A ZERO ?
5467 032206 001401                BEQ   66$ ; BR IF ZERO
5468 032210 104031                ERROR 31 ; REPORT DRIVE IN NEUTRAL
5469 032212 013737 001164 001126 66$: MOV   $TMPO, $BDDAT ; CHECK STATUS FROM PORT A
5470 032220 013737 001224 001240      MOV   PORTA, PTNBR ; CHANGE PORT ADDRESS FOR TYPEOUT
5471 032226 023737 001124 001126      CMP   $GDDAT, $BDDAT ; COMPARE WITH CONSTANT
5472 032234 001401                BEQ   67$ ; BR IF OK
5473 032236 104027                ERROR 27 ; REPORT REGISTER ERROR
5474 032240 000240                NOP
5475 032242 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ; SELECT PORT B
5476 032250 013737 001226 001240      MOV   PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5477 032256 005037 001250                CLR   CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
5478 032262 016037 000012 001126      MOV   RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
5479 032270 012737 000012 001122      MOV   #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
5480 032 16 060037 001122                ADD   RO, $BDAOR ; ADD RHI1 BASE ADDRESS
5481 03 2 02 005037 001124                CLR   $GDDAT ; WHAT REGISTER SHOULD BE
5482 032306 013737 001126 001164      MOV   $BDDAT, $TMPO ; MOVE REGISTER CONTENTS TO '$TMPO'
5483 032314 042737 077777 001164      BIC   #1CATA, $TMPO ; SAVE SPECIFIED BITS
5484 032322 023737 001124 001164      CMP   $GDDAT, $TMPO ; COMPARE THE BITS
5485 032330 001414                BEQ   68$ ; BR IF OK

```


MO8

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 103
T24 SET PORT 'A' REQUEST TEST

SEQ 0103

```

5486 032332 013737 001126 001174      MOV      $BDDAT,$STMP4      ;COPY 'BAD DATA'
5487 032340 042737 100000 001174      BIC      #ATA,$STMP4      ;CLEAR THE MASKED BITS
5488 032346 053737 001174 001124      BIS      $STMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5489 032354 104016          ERROR      16              ;TYPE MESSAGE 16
5490 032356 005137 001250          COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
5491 032362 000240          NOP
5492 032364 113760 001224 000010      68$:     MOVB     PORTA, RMCS2(RO) ;SELECT PORT A
5493 032372 013737 001224 001240      MOV      PORTA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5494 032400 005037 001250          CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
5495 032404 016037 000012 001126      MOV      RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
5496 032412 012737 000012 001122      MOV      #RMDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5497 032420 060037 001122          ADD     RO,$B0ADR        ;ADD RHI1 BASE ADDRESS
5498 032424 012737 100000 001124      MOV      #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
5499 032432 013737 001126 001164      MOV      $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
5500 032440 042737 077777 001164      BIC      #ICATA,$STMP0   ;SAVE SPECIFIED BITS
5501 032446 023737 001124 001164      CMP      $GDDAT,$STMP0   ;COMPARE THE BITS
5502 032454 001414          BEQ     70$              ;BR IF OK
5503 032456 013737 001126 001174      MOV      $BDDAT,$STMP4   ;COPY 'BAD DATA'
5504 032464 042737 100000 001174      BIC      #ATA,$STMP4     ;CLEAR THE MASKED BITS
5505 032472 053737 001174 001124      BIS      $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
5506 032500 104016          ERROR      16              ;TYPE MESSAGE 16
5507 032502 005137 001250          COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
5508 032506 000240          NOP
5509
5510
5511
5512
5513
5514 032510 113760 001224 000010      MOVB     PORTA, RMCS2(RO) ;SELECT PORT A
5515 032516 013737 001224 001240      MOV      PORTA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5516 032524 012760 000013 000000      MOV      #13, RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT A
5517
5518
5519
5520
5521 032532 005037 001254          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
5522 032536 012737 000012 001122      MOV      #RMDS1,$B0ADR    ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5523 032544 060037 001122          ADD     RO,$B0ADR        ;ADD THE I/O BASE ADDRESS
5524 032550 012737 011700 001124      MOV      #MOL!PGH!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5525 032556 113760 001224 000010      MOVB     PORTA, RMCS2(RO) ;SELECT PORT A.
5526 032564 016037 000012 001170      MOV      RMDS1(RO), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5527 032572 042737 024001 001170      BIC      #PIP!WRL!OM,$STMP2 ;CLEAR DONT CARES
5528 032600 013737 001170 001164      MOV      $STMP2,$STMP0   ;COPY IT INTO '$STMP0'
5529 032606 042737 100100 001164      BIC      #ATA!VV,$STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5530 032614 113760 001226 000010      MOVB     PORTB, RMCS2(RO) ;SELECT PORT B.
5531 032622 016037 000012 001172      MOV      RMDS1(RO), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5532 032630 042737 024001 001172      BIC      #PIP!WRL!OM,$STMP3 ;CLEAR DONT CARES
5533 032636 013737 001172 001166      MOV      $STMP3,$STMP1   ;COPY IT INTO '$STMP1'
5534 032644 042737 100100 001166      BIC      #ATA!VV,$STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5535 032652 023737 001164 001166      CMP      $STMP0,$STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
5536 032660 001006          BNE     72$              ;BR IF NOT
5537 032662 005737 001164          TST     $STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO
5538 032666 001045          BNE     74$              ;BR IF NOT
5539 032670 104046          ERROR      46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5540 032672 000137 033056          JMP     76$              ;BYPASS THE REST OF THE CHECKS
5541 032676 013737 001170 001126      72$:     MOV      $STMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5541 032704 013737 001226 001240      MOV      PORTB, PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

;;*****

;RELEASE THE DRIVE FROM PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
REGISTERS ARE THE SAME: ARE THEY ZERO
BR IF NOT
BR IF NOT
REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
BYPASS THE REST OF THE CHECKS
SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

N08

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 104
T24 SET PORT 'A' REQUEST TEST

SEQ 0104

```

5542 032712 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ; SELECT PORT B.
5543 032720 005737 001164              TST    $TMP0             ; SEE IF STATUS EQ 0 FROM PORT A.
5544 032724 001414 73$              BEQ    73$               ; BR IF ZERO
5545 032726 013737 001224 001240      MOV    PORTA, PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5546 032734 013737 001172 001126      MOV    $TMP3, $BDDAT   ; 'BAD DATA' FOR ERROR TYPE OUT
5547 032742 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ; SELECT PORT A.
5548 032750 005737 001166              TST    $TMP1             ; SEE IF STATUS EQ ZERO FROM PORT B.
5549 032754 001012 74$              BNE    74$               ; BR IF NOT
5550 032756 012737 177777 001254 73$:      MOV    #-1, RELEARR    ; SET 'RELEASE ERROR' INDICATOR
5551 032764 012760 000011 000000      MOV    #11, RMCS1(RO)  ; CLEAR THE DRIVE
5552 032772 012760 000013 000000      MOV    #13, RMCS1(RO)  ; RELEASE THE DRIVE
5553 033000 104026 74$              ERROR  26               ; TYPE ERROR MESSAGE 26
5554 033002 013737 001170 001126 74$:      MOV    $TMP2, $BDDAT   ; LOOK FOR BIT FAILURES WHEN RMDS1 READ
5555 033010 013737 001224 001240      MOV    PORTA, PTNBR     ; CHANGE PORT NUMBER
5556 033016 023737 001124 001126      CMP    $GDDAT, $GDDAT  ; ALL BITS OK ?
5557 033024 001401 75$              BEQ    75$               ; BR IF OK FROM PORT A.
5558 033026 104007 75$              ERROR  7                ; REPORT ERROR
5559 033030 013737 001172 001126 75$:      MOV    $TMP3, $BDDAT   ; CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5560 033036 013737 001226 001240      MOV    PORTB, PTNBR     ; CHANGE PORT NUMBER
5561 033044 023737 001124 001126      CMP    $GDDAT, $BDDAT  ; SEE IF READ OK FROM PORT B.
5562 033052 001401 76$              BEQ    76$               ; BR IF OK
5563 033054 104007 76$              ERROR  7                ; REPORT ERROR
5564 033056 000240 1$:              NOP
5565 033060 000004 1$:              SCOPE                   ; LOOP ?

```

```

5566
5567 *****
5568 *TEST 25      SET PORT 'B' REQUEST TEST
5569 *
5570 *VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
5571 *DRIVE IS SEIZED BY THE OTHER PORT.
5572 *
5573 * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
5574 *
5575 * B. WRITE 0'S INTO RMDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
5576 *SEIZED BY PORT 'A'.
5577 *
5578 * C. ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
5579 *SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR
5580 *PORT 'B' AND IS NOT SET FOR PORT 'A'.
5581 *
5582 * D. ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
5583 *RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5584 *
5585 *****

```

```

5586 033062 005737 001300      †ST25: TST    KYBCTL           ; PERFORMING ONLY SINGLE TESTS ?
5587 033062 001406 1$              BEQ    2$                ; BR IF NOT
5588 033066 100002 1$              BPL    1$                ; BR IF JUST ENTERED TEST
5589 033070 000137 002676 001300 1$:      JMP    EXEC              ; RETURN & GET NEXT TEST NUMBER
5590 033072 012737 177777 001300 25:      MOV    #-1, KYBCTL      ; SET SINGLE TEST INDICATOR
5591 033076 112737 000025 001102 25:      MOVB   #25, $STSTNM     ; TEST NUMBER
5592 033104 012737 033134 001106      MOV    #TEST25, $LPADR  ; LOAD LOOP ON TEST ADDRESS
5593 033112 012737 033134 001110      MOV    #TEST25, $LPERR  ; LOAD LOOP ON ERROR ADDRESS
5594 033120 012737 000031 001176      MOV    #25, $TIMES      ; DO 25 ITERATIONS
5595 033126 012706 001100 25:      MOV    $STACK, SP      ; LOAD THE STACK POINTER
5596 033134
5597

```

```

5598 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5599
5600 033140 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT #A
5601 033146 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
5602 033152 012760 000011 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
5603 033160 012760 000013 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5604 033166 113760 001226 MOVB PORTB, RMCS2(RO) ;SELECT PORT #B
5605 033174 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
5606 033200 012760 000011 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
5607 033206 012760 000013 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5608
5609 ;*****
5610
5611 ;SEIZE THE DRIVE THROUGH PORT A
5612
5613 033214 113760 001224 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5614 033222 013737 001224 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
5615 033230 005060 000012 CLR RMDS1(RO) ;WRITE RMDS1
5616 033234 013737 001226 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
5617 033242 113760 001226 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
5618 033250 013737 001226 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5619
5620 ;*****
5621 ;SET PORT REQUEST
5622
5623 033256 005060 000012 CLR RMDS1(RO) ;SET PORT REQUEST FOR PORT B
5624
5625 ;*****
5626 ;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.
5627
5628
5629 ;RELEASE THE DRIVE FROM PORT A
5630
5631 033262 113760 001224 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5632 033270 013737 001224 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5633 033276 012760 000013 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5634
5635 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5636
5637 033304 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5638 033310 012737 111700 MOV #ATA:MOL:PGM:DPR:DRY:VV, $GDDAT ;COMPARISON CONSTANT
5639 033316 012737 000012 MOV #RMDS1, $BDADR ;REGISTER ADDRESS INCREMENT
5640 033324 060037 001122 ADD RO, $BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
5641 033330 113760 001226 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
5642 033336 013737 001226 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5643 033344 016037 000012 MOV RMDS1(RO), $TMPD ;READ STATUS REGISTER FROM PORT B
5644 033352 113760 001224 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5645 033360 013737 001224 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5646 033366 016037 000012 MOV RMDS1(RO), $BDADR ;DRIVE STATUS FROM PORT A
5647 033374 001404 BEQ 66$ ;BR IF STATUS FROM PORT A ZERO
5648 033376 005737 001164 TST $TMPD ;IS STATUS FROM PORT B ZERO?
5649 033402 001401 BEQ 66$ ;BR IF ZERO
5650 033404 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
5651 033406 013737 001164 66$: MOV $TMPD, $BDADR ;CHECK STATUS FROM PORT B
5652 033414 013737 001226 MOV PORTB, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5653 033422 023737 001124 CMP $GDDAT, $BDADR ;COMPARE WITH CONSTANT

```

```

5654 033430 001401 BEQ 67$ ;BR IF OK
5655 033432 104027 ERROR 27 ;REPORT REGISTER ERROR
5656 033434 000240 NOP
5657 033436 113760 001224 000010 67$: MOVB PORTA, RMCS2(RO) ;SELECT PORT A
5658 033444 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5659 033452 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5660 033456 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
5661 033464 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5662 033472 060037 001122 ADD RO, $B0ADR ;ADD RHI1 BASE ADDRESS
5663 033476 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5664 033502 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5665 033510 042737 077777 001164 BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
5666 033516 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
5667 033524 001414 BEQ 68$ ;BR IF OK
5668 033526 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
5669 033534 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
5670 033542 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5671 033550 104016 ERROR 16 ;TYPE MESSAGE 16
5672 033552 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5673 033556 000240 NOP
5674 033560 113760 001226 000010 68$: MOVB PORTB, RMCS2(RO) ;SELECT PORT B
5675 033566 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5676 033574 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5677 033600 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
5678 033606 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5679 033614 060037 001122 ADD RO, $B0ADR ;ADD RHI1 BASE ADDRESS
5680 033620 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
5681 033626 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5682 033634 042737 077777 001164 BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
5683 033642 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
5684 033650 001414 BEQ 70$ ;BR IF OK
5685 033652 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
5686 033660 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
5687 033666 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5688 033674 104016 ERROR 16 ;TYPE MESSAGE 16
5689 033676 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5690 033702 000240 NOP
5691
5692 ;:*****
5693
5694 ;RELEASE THE DRIVE FROM PORT B
5695
5696 033704 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
5697 033712 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5698 033720 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5699
5700 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5701
5702 033726 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5703 033732 012737 000012 001122 MOV #RMDS1, $B0ADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5704 033740 060037 001122 ADD RO, $B0ADR ;ADD THE I/O BASE ADDRESS
5705 033744 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
5706 033752 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
5707 033760 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5708 033766 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
5709 033774 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'

```

```

5710 034002 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5711 034010 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
5712 034016 016037 000012 001172 MOV RMDS1(RO), STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5713 034024 042737 024001 001172 BIC #PIP!WAL!OM, STMP3 ;CLEAR DONT CARES
5714 034032 013737 001172 001166 MOV STMP3, STMP1 ;COPY IT INTO 'STMP1'
5715 034040 042737 100100 001166 BIC #ATA!VV, STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5716 034046 023737 001164 001166 CMP STMP0, STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5717 034054 001006 BNE 72$ ;BR IF NOT
5718 034056 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5719 034062 001045 BNE 74$ ;BR IF NOT
5720 034064 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5721 034066 000137 034252 JMP 76$ ;BYPASS THE REST OF THE CHECKS
5722 034072 013737 001170 001126 72$: MOV STMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5723 034100 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5724 034106 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
5725 034114 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5726 034120 001414 BEQ 73$ ;BR IF ZERO
5727 034122 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5728 034130 013737 001172 001126 MOV STMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5729 034136 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
5730 034144 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5731 034150 001012 BNE 74$ ;BR IF NOT
5732 034152 012737 177777 001254 73$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
5733 034160 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
5734 034166 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
5735 034174 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5736 034176 013737 001170 001126 74$: MOV STMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5737 034204 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
5738 034212 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
5739 034220 001401 BEQ 75$ ;BR IF OK FROM PORT A.
5740 034222 104007 ERROR 7 ;REPORT ERROR
5741 034224 013737 001172 001126 75$: MOV STMP3, $BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5742 034232 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
5743 034240 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
5744 034246 001401 BEQ 76$ ;BR IF OK
5745 034250 104007 ERROR 7 ;REPORT ERROR
5746 034252 000240 76$: NOP
5747 034254 1$: SCOPE ;LOOP ?

```

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

```

*****
*TEST 26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR
*
*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
* SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
*
* C. ISSUE A DRIVE CLEAR COMMAND.
*
* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION

```

```

5766
5767
5768
5769
5770 034256
5771 034256 005737 001300
5772 034262 001406
5773 034264 100002
5774 034266 000137 002676
5775 034272 012737 177777 001300
5776 034300 112737 000026 001102
5777 034306 012737 034330 001106
5778 034314 012737 034330 001110
5779 034322 012737 000031 001176
5780 034330 012706 001100
5781
5782
5783
5784
5785 034334 113760 001224 000010
5786 034342 005760 000012
5787 034346 001775
5788 034350 012760 177777 000014
5789 034356 005060 000014
5790 034362 013760 001226 000010
5791 034370 005760 000012
5792 034374 001775
5793 034376 012760 177777 000014
5794 034404 005060 000014
5795 034410 113760 001224 000010
5796 034416 005760 000012
5797 034422 001775
5798
5799
5800
5801
5802 034424 113760 001224 000010
5803 034432 013737 001224 001240
5804 034440 005037 001250
5805 034444 016037 000012 001126
5806 034452 012737 000012 001122
5807 034460 060037 001122
5808 034464 012737 100000 001124
5809 034472 013737 001126 001164
5810 034500 042737 077777 001164
5811 034506 023737 001124 001164
5812 034514 001414
5813 034516 013737 001126 001174
5814 034524 042737 100000 001174
5815 034532 053737 001174 001124
5816 034540 104010
5817 034542 005137 001250
5818 034546 000240
5819 034550 005737 001250
5820 034554 001402
5821 034556 000137 035750

```

```

;* BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
;* 'B' IS STILL SET.
*****
↑ST26:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #26,$STSTNM ;TEST NUMBER
MOV #TEST26,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST26,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST26: MOV #STACK,SP ;LOAD THE STACK POINTER
;*****
;SET ATTENTION BITS FOR BOTH PORTS
66$: MOVB PORTA, RMCS2(RO) ;SELECT PORT 64$
TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
BEQ 66$
MOV #-1, RMER1(RO) ;FORCE ERRORS
CLR RMER1(RO) ;CLEAR THE ERRORS
64$: MOV PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1, RMER1(RO) ;FORCE ERRORS ON PORT 65$
CLR RMER1(RO) ;CLEAR THE ERRORS
65$: MOVB PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
MOV #RMD51, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAOR ;ADD RHI1 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 67$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
67$: NOP
TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
BEQ +6 ;BR IF IT WAS
JMP 15 ;BYPASS REST OF TEST IF NOT

```

F09

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 109
T26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR

SEQ 0109

5822	034562	113760	001226	000010	MOV B	PORTB, RMCS2(RO)	; SELECT PORT B
5823	034570	013737	001226	001240	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5824	034576	005037	001250		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
5825	034602	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; GET CONTENTS OF RMDS1
5826	034610	012737	000012	001122	MOV	#RMDS1, \$BDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
5827	034616	060037	001122		ADD	RO, \$BDADR	; ADD RH11 BASE ADDRESS
5828	034622	012737	100000	001124	MOV	#ATA, \$GDDAT	; WHAT REGISTER SHOULD BE
5829	034630	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'
5830	034636	042737	077777	001164	BIC	#!CATA, \$TMP0	; SAVE SPECIFIED BITS
5831	034644	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
5832	034652	001414			BEQ	69\$; BR IF OK
5833	034654	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
5834	034662	042737	100000	001174	BIC	#ATA, \$TMP4	; CLEAR THE MASKED BITS
5835	034670	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
5836	034676	104010			ERROR	10	; REPORT THE ERROR
5837	034700	005137	001250		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
5838	034704	000240			69\$:	NOP	
5839	034706	005737	001250		TST	CKERR	; WAS ATTN BIT FOR PORT B SET ?
5840	034712	001402			BEQ	+6	; BR IF IT WAS
5841	034714	000137	035750		JMP	1\$; BYPASS REST OF TEST IF NOT

; *****

; SEIZE THE DRIVE THROUGH PORT A

5847	034720	113760	001224	000010	MOV B	PORTA, RMCS2(RO)	; SELECT PORT A
5848	034726	013737	001224	001242	MOV	PORTA, SEIZPT	; STORE SEIZING PORT'S ADDRESS
5849	034734	005060	000012		CLR	RMDS1(RO)	; WRITE RMDS1
5850	034740	113760	001226	000010	MOV B	PORTB, RMCS2(RO)	; SELECT PORT B
5851	034746	013737	001226	001240	MOV	PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5852	034754	013737	001226	001244	MOV	PORTB, OPPRT	; 'OPPOSITE' PORT ADDRESS
5853	034762	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; SEE IF DRIVE SEIZED BY PORT A
5854	034770	010037	001122		MOV	RO, \$BDADR	; RH11 BASE ADDRESS
5855	034774	062737	000012	001122	ADD	#RMDS1, \$BDADR	; GENERATE BAD REGISTER ADDRESS
5856	035002	005037	001124		CLR	\$GDDAT	; REGISTER SHOULD BE ZERO
5857	035006	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	; IS THE REGISTER ZERO
5858	035014	001403			BFC	71\$; BR IF IT IS
5859	035016	104004			ERROR	4	; REPORT THE ERROR
5860	035020	000137	035750		JMP	1\$; BYPASS REST OF THE SUBTEST
5861	035024				71\$:		
5862	035024	113760	001224	000010	MOV B	PORTA, RMCS2(RO)	; SELECT PORT A
5863	035032	013737	001224	001240	MOV	PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5864	035040	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	; SEE IF SEIZING PORT SEES CORRECT STATUS
5865	035046	042737	020001	001126	BIC	#OM!PIP, \$BDDAT	; CLEAR DONT CARE BITS
5866	035054	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	; EXPECTED STATUS
5867	035062	013737	001124	001166	MOV	\$GDDAT, \$TMP1	; USE GOOD DATA AS A MASK
5868	035070	005137	001166		COM	\$TMP1	; COMPLEMENT THE EXPECTED STATUS
5869	035074	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; SAVE THE ACTUAL STATUS
5870	035102	043737	001166	001164	BIC	\$TMP1, \$TMP0	; CLEAR UNWANTED BITS
5871	035110	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; ARE THE EXPECTED STATUS BITS SET ?
5872	035116	001401			BEQ	72\$; BR IF THEY ARE
5873	035120	104005			ERROR	5	; REPORT THE ERROR
5874	035122	000240			72\$:	NOP	

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

```

5878
5879 035124 012760 000011 000000          MOV      #11,RMCS1(RO) ;DO A DRIVE CLEAR COMMAND
5880
5881      ;*****
5882      ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
5883
5884 035132 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
5885 035136 016037 000012 001126          MOV      RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
5886 035144 012737 000012 001122          MOV      #RMDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5887 035152 060037 001122          ADD      RO,SBDAOR     ;ADD R#11 BASE ADDRESS
5888 035156 005037 001124          CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
5889 035162 013737 001126 001164          MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
5890 035170 042737 077777 001164          BIC      #1CATA,$TMP0 ;SAVE SPECIFIED BITS
5891 035176 023737 001124 001164          CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
5892 035204 001414          BEQ      73$          ;BR IF OK
5893 035206 013737 001126 001174          MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
5894 035214 042737 100000 001174          BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
5895 035222 053737 001174 001124          BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5896 035230 104047          ERROR   47          ;TYPE MESSAGE 47
5897 035232 005137 001250          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
5898 035236 000240          73$: NOP
5899
5900      ;*****
5901
5902      ;RELEASE THE DRIVE FROM PORT A
5903
5904 035240 113760 001224 000010          MOV      PORTA,RMCS2(RO) ;SELECT PORT A
5905 035246 013737 001224 001240          MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5906 035254 012760 000013 000000          MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5907
5908      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5909
5910 035262 005037 001254          CLR      RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
5911 035266 012737 000012 001122          MOV      #RMDS1,SBDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
5912 035274 060037 001122          ADD      RO,SBDAOR     ;ADD THE I/O BASE ADDRESS
5913 035300 012737 011700 001124          MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5914 035306 113760 001224 000010          MOV      PORTA,RMCS2(RO) ;SELECT PORT A
5915 035314 016037 000012 001170          MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
5916 035322 042737 024001 001170          BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
5917 035330 013737 001170 001164          MOV      $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
5918 035336 042737 100100 001164          BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5919 035344 113760 001226 000010          MOV      PORTB,RMCS2(RO) ;SELECT PORT B
5920 035352 016037 000012 001172          MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B
5921 035360 042737 024001 001172          BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
5922 035366 013737 001172 001166          MOV      $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
5923 035374 042737 100100 001166          BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5924 035402 023737 001164 001166          CMP      $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ^
5925 035410 001006          BNE      75$          ;BR IF NOT
5926 035412 005737 001164          TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ^
5927 035416 001045          BNE      77$          ;BR IF NOT
5928 035420 104046          ERROR   46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5929 035422 000137 035622          JMP      79$          ;BYPASS THE REST OF THE CHECKS
5930 035426 013737 001170 001126 75$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5931 035434 013737 001226 001240          MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5932 035442 113760 001226 000010          MOV      PORTB,RMCS2(RO) ;SELECT PORT B
5933 035450 005737 001164          TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.

```



```

5934 035454 001414 BEQ 76$ ;BR IF ZERO
5935 035456 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5936 035464 013737 001172 001126 STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5937 035472 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
5938 035500 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5939 035504 001012 BNE 77$ ;BR IF NOT
5940 035506 012737 177777 001254 76$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
5941 035514 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
5942 035522 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
5943 035530 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5944 035532 013737 001170 001126 77$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
5945 035540 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5946 035546 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5947 035554 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
5948 035562 001401 BEQ 78$ ;BR IF OK FROM PORT A.
5949 035564 104007 ERROR 7 ;REPORT ERROR
5950 035566 013737 001172 001126 78$: MOV STMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
5951 035574 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5952 035602 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
5953 035610 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
5954 035616 001401 BEQ 79$ ;BR IF OK
5955 035620 104007 ERROR 7 ;REPORT ERROR
5956 035622 000240 79$: NOP

```

```

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

```

```

5961 035624 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
5962 035632 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5963 035640 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5964 035644 016037 000012 001126 MOV RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
5965 035652 012737 000012 001122 MOV #RMDS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5966 035660 060037 001122 ADD RO,$BADDR ;ADD RHI1 BASE ADDRESS
5967 035664 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
5968 035672 013737 001126 001164 MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
5969 035700 042737 077777 001164 BIC #ICATA,$STMP0 ;SAVE SPECIFIED BITS
5970 035706 023737 001124 001164 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
5971 035714 001414 BEQ 80$ ;BR IF OK
5972 035716 013737 001126 001174 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
5973 035724 042737 100000 001174 BIC #ATA,$STMP4 ;CLEAR THE MASKED BITS
5974 035732 053737 001174 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5975 035740 104050 ERROR 50 ;TYPE MESSAGE 50
5976 035742 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5977 035746 000240 80$: NOP
5978 035750 000004 1$: SCOPE ;LOOP ?

```

```

*****
*TEST 27 TEST RESET ATTENTION 'B' BY DRIVE CLEAR
*
*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
* SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.

```



```

5990
5991
5992
5993
5994
5995
5996
5997
5998 035752
5999 035752 005737 001300
6000 035756 001406
6001 035760 100002
6002 035762 000137 002676
6003 035766 012737 177777 001300
6004 035774 112737 000027 001102
6005 036002 012737 036024 001106
6006 036010 012737 036024 001110
6007 036016 012737 000031 001176
6008 036024 012706 001100
6009
6010
6011
6012
6013 036030 113760 001224 000010
6014 036036 005760 000012
6015 036042 001775
6016 036044 012760 177777 000014
6017 036052 005060 000014
6018 036056 013760 001226 000010
6019 036064 005760 000012
6020 036070 001775
6021 036072 012760 177777 000014
6022 036100 005060 000014
6023 036104 113760 001224 000010
6024 036112 005760 000012
6025 036116 001775
6026
6027
6028
6029
6030 036120 113760 001226 000010
6031 036126 013737 001226 001240
6032 036134 005037 001250
6033 036140 016037 000012 001126
6034 036146 012737 000012 001122
6035 036154 060037 001122
6036 036160 012737 100000 001124
6037 036166 013737 001126 001164
6038 036174 042737 077777 001164
6039 036202 023737 001124 001164
6040 036210 001414
6041 036212 013737 001126 001174
6042 036220 042737 100000 001174
6043 036226 053737 001174 001124
6044 036234 104010
6045 036236 005137 001250

```

```

;*
;* C. ISSUE A DRIVE CLEAR COMMAND.
;*
;* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION
;* BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
;* 'A' IS STILL SET.
;*
*****
↑ST27:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      25         ;BR IF NOT
      BPL      15         ;BR IF JUST ENTERED TEST
      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
1$:   MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
2$:   MOVB     #27,$STNM   ;TEST NUMBER
      MOV      #TEST27,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST27,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #25,$TIMES  ;DO 25 ITERATIONS
TEST27: MOV     #STACK,SP ;LOAD THE STACK POINTER
;*****
;SET ATTENTION BITS FOR BOTH PORTS
66$:  MOVB     PORTA,RMCS2(RO) ;SELECT PORT 64$
      TST     RMDS1(RO)      ;MAKE SURE DRIVE AVAILABLE
      BEQ     66$
      MOV     #-1,RMER1(RO) ;FORCE ERRORS
      CLR     RMER1(RO)     ;CLEAR THE ERRORS
64$:  MOV     PORTB,RMCS2(RO) ;SELECT THE OTHER PORT
      TST     RMDS1(RO)     ;WAIT FOR DRIVE TO TIMEOUT
      BEQ     64$          ;BR IF DRIVE HASN'T TIMED OUT
      MOV     #-1,RMER1(RO) ;FORCE ERRORS ON PORT 65$
      CLR     RMER1(RO)     ;CLEAR THE ERRORS
65$:  MOVB     PORTA,RMCS2(RO) ;SELECT PORT "64$" AGAIN
      TST     RMDS1(RO)     ;WAIT FOR DRIVE TO TIMEOUT
      BEQ     65$          ;BR IF DRIVE HASN'T TIMED OUT
;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
      MOV     RMDS1(RO),%BDDAT ;GET CONTENTS OF RMDS1
      MOV     #RMDS1,%BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      ADD     RO,%BCADR    ;ADD RHI1 BASE ADDRESS
      MOV     #ATA,%GDDAT  ;WHAT REGISTER SHOULD BE
      MOV     %BDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
      BIC     #+CATA,$TMPO ;SAVE SPECIFIED BITS
      CMP     %GDDAT,$TMPO ;COMPARE THE BITS
      BEQ     67$         ;BR IF OK
      MOV     %BDDAT,$TMP4 ;COPY 'BAD DATA'
      BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
      BIS     $TMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
      LD      ERROR      ;REPORT THE ERROR
      COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR

```

```

6046 036242 000240
6047 036244 005737 001250
6048 036250 001402
6049 036252 000137 037444
6050 036256 113760 001224 000010
6051 036264 013737 001224 001240
6052 036272 005037 001250
6053 036276 016037 000012 001126
6054 036304 012737 000012 001122
6055 036312 060037 001122
6056 036316 012737 100000 001124
6057 036324 013737 001126 001164
6058 036332 042737 077777 001164
6059 036340 023737 001124 001164
6060 036346 001414
6061 036350 013737 001126 001174
6062 036356 042737 100000 001174
6063 036364 053737 001174 001124
6064 036372 104010
6065 036374 005137 001250
6066 036400 000240
6067 036402 005737 001250
6068 036406 001402
6069 036410 000137 037444
6070
6071
6072
6073
6074
6075 036414 113760 001226 000010
6076 036422 013737 001226 001242
6077 036430 005060 000012
6078 036434 113760 001224 000010
6079 036442 013737 001224 001240
6080 036450 013737 001224 001244
6081 036456 016037 000012 001126
6082 036464 010037 001122
6083 036470 062737 000012 001122
6084 036476 005037 001124
6085 036502 023737 001124 001126
6086 036510 001403
6087 036512 104004
6088 036514 000137 037444
6089 036520
6090 036520 113760 001226 000010
6091 036526 013737 001226 001240
6092 036534 016037 000012 001126
6093 036542 042737 020001 001126
6094 036550 012737 011700 001124
6095 036556 013737 001124 001166
6096 036564 005137 001166
6097 036570 013737 001126 001164
6098 036576 043737 001166 001164
6099 036604 023737 001124 001164
6100 036612 001401
6101 036614 104005

```

```

67$: NOP
TST CKERR ; WAS ATTN BIT FOR PORT B SET ?
BEQ .+6 ; BR IF IT WAS
JMP 1$ ; BYPASS REST OF TEST IF NOT
MOV#B PORTA, RMCS2(RO) ; SELECT PORT A
MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
MOV #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAOR ; ADD RH11 BASE ADDRESS
MOV #ATA, $GDDAT ; WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
BIC #1CATA, $TMP0 ; SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ; COMPARE THE BITS
BEQ 69$ ; 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

69$: NOP
TST CKERR ; WAS ATTN BIT FOR PORT A SET ?
BEQ .+6 ; BR IF IT WAS
JMP 1$ ; BYPASS REST OF TEST IF NOT

;*****
; SEIZE THE DRIVE THROUGH PORT B
MOV#B PORTB, RMCS2(RO) ; SELECT PORT B
MOV PORTB, SEIZPT ; STORE SEIZING PORT'S ADDRESS
CLR RMDS1(RO) ; WRITE RMDS1
MOV#B PORTA, RMCS2(RO) ; SELECT PORT A
MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA, OPPRT ; 'OPPOSITE' PORT ADDRESS
MOV RMDS1(RO), $BDDAT ; SEE IF DRIVE SEIZED BY PORT B
MOV RO, $BDAOR ; RH11 BASE ADDRESS
ADD #RMDS1, $BDAOR ; GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ; REGISTER SHOULD BE ZERO
CMP $GDDAT, $BDDAT ; IS THE REGISTER ZERO
BEQ 71$ ; BR IF IT IS
ERROR 4 ; REPORT THE ERROR
JMP 1$ ; BYPASS REST OF THE SUBTEST

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

```

```

6102 036616 000240          72$:  NOP
6103
6104
6105          ;*****
6106          ;ISSUE DRIVE CLEAR COMMAND TO PORT B
6107 036620 012760 000011 000000          MOV      #11,RMCS1(RO) ;DO A DRIVE CLEAR COMMAND
6108
6109          ;*****
6110          ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
6111
6112 036626 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
6113 036632 016037 000012 001126          MOV      RMDS1(RO),SBDAT ;GET CONTENTS OF RMDS1
6114 036640 012737 000012 001122          MOV      #RMDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6115 036646 060037 001122          ADD      RO,SBDADR      ;ADD RM11 BASE ADDRESS
6116 036652 005037 001124          CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
6117 036656 013737 001126 001164          MOV      SBDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6118 036664 042737 077777 001164          BIC      #!CAT,$TMP0    ;SAVE SPECIFIED BITS
6119 036672 023737 001124 001164          CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
6120 036700 001414          BEQ      73$           ;BR IF OK
6121 036702 013737 001126 001174          MOV      SBDAT,$TMP4    ;COPY 'BAD DATA'
6122 036710 042737 100000 001174          BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
6123 036716 053737 001174 001124          BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
6124 036724 104047          ERROR   47           ;TYPE MESSAGE 47
6125 036726 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
6126 036732 000240          73$:  NOP
6127
6128          ;*****
6129
6130          ;RELEASE THE DRIVE FROM PORT B
6131
6132 036734 113760 001226 000010          MOV      PORTB,RMCS2(RO) ;SELECT PORT B
6133 036742 013737 001226 001240          MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6134 036750 012760 000013 000000          MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
6135
6136          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6137
6138 036756 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
6139 036762 012737 000012 001122          MOV      #RMDS1,SBDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
6140 036770 060037 001122          ADD      RO,SBDADR      ;ADD THE I/O BASE ADDRESS
6141 036774 012737 011700 001124          MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6142 037002 113760 001224 000010          MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
6143 037010 016037 000012 001170          MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6144 037016 042737 024001 001170          BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
6145 037024 013737 001170 001164          MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
6146 037032 042737 100100 001164          BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6147 037040 113760 001226 000010          MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
6148 037046 016037 000012 001172          MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6149 037054 042737 024001 001172          BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
6150 037062 013737 001172 001166          MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
6151 037070 042737 100100 001166          BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6152 037076 023737 001164 001166          CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6153 037104 001006          BNE      75$           ;BR IF NOT
6154 037106 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6155 037112 001045          BNE      77$           ;BR IF NOT
6156 037114 104046          ERROR   46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6157 037116 000137 037316          JMP      79$           ;BYPASS THE REST OF THE CHECKS

```

```

6156 037122 013737 001170 001126 75$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6159 037130 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6160 037136 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
6161 037144 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6162 037150 001414 BEQ 76$ ;BR IF ZERO
6163 037152 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6164 037160 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6165 037166 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
6166 037174 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6167 037200 001012 BNE 77$ ;BR IF NOT
6168 037202 012737 177777 001254 76$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6169 037210 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
6170 037216 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
6171 037224 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6172 037226 013737 001170 001126 77$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6173 037234 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6174 037242 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6175 037250 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
6176 037256 001401 BEQ 78$ ;BR IF OK FROM PORT A.
6177 037260 104007 ERROR 7 ;REPORT ERROR
6178 037262 013737 001172 001126 78$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6179 037270 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6180 037276 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6181 037304 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6182 037312 001401 BEQ 79$ ;BR IF OK
6183 037314 104007 ERROR 7 ;REPORT ERROR
6184 037316 000240 79$: NOP
6185
6186
6187
6188

```

```

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

```

```

6189 037320 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
6190 037326 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6191 037334 005037 001250 001240 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6192 037340 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
6193 037346 012737 000012 001122 MOV #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6194 037354 060037 001122 001124 ADD RO,$BDDADR ;ADD RHI1 BASE ADDRESS
6195 037360 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6196 037366 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6197 037374 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
6198 037402 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6199 037410 001414 BEQ 80$ ;BR IF OK
6200 037412 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6201 037420 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6202 037426 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6203 037434 104050 ERROR 50 ;TYPE MESSAGE 50
6204 037436 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6205 037442 000240 80$: NOP
6206 037444 000004 1$: SCOPE ;LOOP ?
6207
6208
6209
6210
6211
6212
6213

```

```

*****
;TEST 30 RESET ATTENTION 'A' BY GO TEST
;
;* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE

```

```

6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229 037446
6230 037446 005737 001300
6231 037452 001406
6232 037454 100002
6233 037456 000137 002676
6234 037462 012737 177777 001300
6235 037470 112737 000030 001102
6236 037476 012737 037520 001106
6237 037504 012737 037520 001110
6238 037512 012737 000031 001176
6239 037520 012706 001100
6240
6241
6242
6243
6244
6245 037524 113760 001224 000010
6246 037532 005760 000012
6247 037536 001775
6248 037540 012760 177777 000014
6249 037546 005060 000014
6250 037552 013760 001226 000010
6251 037560 005760 000012
6252 037564 001775
6253 037566 012760 177777 000014
6254 037574 005060 000014
6255 037600 113760 001224 000010
6256 037606 005760 000012
6257 037612 001775
6258
6259
6260
6261
6262 037614 113760 001224 000010
6263 037622 013737 001224 001240
6264 037630 005037 001250
6265 037634 016037 000012 001126
6266 037642 012737 000012 001122
6267 037650 060037 001122
6268 037654 012737 100000 001124
6269 037662 013737 001126 001164

```

```

;* SEIZING PORT.
;*
;* A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
;* ATTENTION BITS ARE SET.
;*
;* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S
;* INTO RMD5.
;*
;* C. ISSUE A NOP COMMAND.
;*
;* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE
;* ATTENTION BIT FOR PORT 'A' IS RESET, AND THE
;* ATTENTION BIT FOR PORT 'B' IS STILL SET.
*****
T30:
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 #30, $TSTNM ;TEST NUMBER
MOV #TEST30, $LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST30, $LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25, $TIMES ;DO 25 ITERATIONS
TEST30: MOV #STACK, SP ;LOAD THE STACK POINTER
*****
;SET ATTENTION BITS FOR BOTH PORTS
66$: MOVB PORTA, RMCS2(R0) ;SELECT PORT 64$
TST RMD51(R0) ;MAKE SURE DRIVE AVAILABLE
BEQ 66$
MOV #-1, RMER1(R0) ;FORCE ERRORS
CLR RMER1(R0) ;CLEAR THE ERRORS
64$: MOV PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
TST RMD51(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1, RMER1(R0) ;FORCE ERRORS ON PORT 65$
CLR RMER1(R0) ;CLEAR THE ERRORS
65$: MOVB PORTA, RMCS2(R0) ;SELECT PORT "64$" AGAIN
TST RMD51(R0) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA, RMCS2(R0) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD51(R0), $BDDAT ;GET CONTENTS OF RMD51
MOV #RMD51, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDAOR ;ADD RHI1 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'

```

6270	037670	042737	077777	001164	BIC	#1CATA,\$TMP0	;SAVE SPECIFIED BITS
6271	037676	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
6272	037704	001414			BEQ	67\$;BR IF OK
6273	037706	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
6274	037714	042737	100000	001174	BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
6275	037722	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
6276	037730	104010			ERROR	10	;REPORT THE ERROR
6277	037732	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6278	037736	000240			NOP		
6279	037740	005737	001250		TST	CKERR	;WAS ATTENTION SET FOR A??
6280	037744	001402			BEQ	+6	;YES!!
6281	037746	000137	041140		JMP	1\$;NO - BYPASS REST OF TEST
6282	037752	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
6283	037760	013737	001226	001240	MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6284	037766	005037	001250		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
6285	037772	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
6286	040000	012737	000012	001122	MOV	#RMDS1, \$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
6287	040006	060037	001122		ADD	RU, \$BDAOR	;ADD RHI1 BASE ADDRESS
6288	040012	012737	100000	001124	MOV	#ATA, \$GDDAT	;WHAT REGISTER SHOULD BE
6289	040020	013737	001126	001164	MOV	\$BDDAT, \$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
6290	040026	042737	077777	001164	BIC	#1CATA,\$TMP0	;SAVE SPECIFIED BITS
6291	040034	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
6292	040042	001414			BEQ	69\$;BR IF OK
6293	040044	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
6294	040052	042737	100000	001174	BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
6295	040060	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
6296	040066	104010			ERROR	10	;REPORT THE ERROR
6297	040070	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
6298	040074	000240			NOP		
6299	040076	005737	001250		TST	CKERR	;WAS ATTENTION SET FOR B??
6300	040102	001402			BEQ	+6	;YES!!
6301	040104	000137	041140		JMP	1\$;NO - BYPASS REST OF TEST

67\$:

69\$:

;;*****

;SEIZE THE DRIVE THROUGH PORT A

6302							
6303							
6304							
6305							
6306							
6307	040110	113760	001224	000010	MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
6308	040116	013737	001224	001242	MOV	PORTA, SEIZPT	;STORE SEIZING PORT'S ADDRESS
6309	040124	005060	000012		CLR	RMDS1(RO)	;WRITE RMDS1
6310	040130	113760	001226	000010	MOVB	PORTB, RMCS2(RO)	;SELECT PORT B
6311	040136	013737	001226	001240	MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6312	040144	013737	001226	001244	MOV	PORTB, OPPRT	; 'OPPOSITE' PORT ADDRESS
6313	040152	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	;SEE IF DRIVE SEIZED BY PORT A
6314	040160	010037	001122		MOV	RO, \$BDAOR	;RHI1 BASE ADDRESS
6315	040164	062737	000012	001122	ADD	#RMDS1, \$BDAOR	;GENERATE BAD REGISTER ADDRESS
6316	040172	005037	001124		CLR	\$GDDAT	;REGISTER SHOULD BE ZERO
6317	040176	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	;IS THE REGISTER ZERO
6318	040204	001403			BEQ	71\$;BR IF IT IS
6319	040206	104004			ERROR	4	;REPORT THE ERROR
6320	040210	000137	041140		JMP	1\$;BYPASS REST OF THE SUBTEST
6321	040214						
6322	040214	113760	001224	000010	MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
6323	040222	013737	001224	001240	MOV	PORTA, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6324	040230	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	;SEE IF SEIZING PORT SEES CORRECT STATUS
6325	040236	042737	020001	001126	BIC	#OM!PIP, \$BDDAT	;CLEAR DONT CARE BITS

71\$:

```

6326 040244 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
6327 040252 013737 001124 001166      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
6328 040260 005137 001166      COM      $TMP1 ;COMPLEMENT THE EXPECTED STATUS
6329 040264 013737 001126 001164      MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
6330 040272 043737 001166 001164      BIC      $TMP1,$TMP0 ;CLEAR UNWANTED BITS
6331 040300 023737 001124 001164      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
6332 040306 001401      BEQ      72$ ;BR IF THEY ARE
6333 040310 104005      ERROR 5 ;REPORT THE ERROR
6334 040312 000240      72$: NOP
6335
6336 ;:*****
6337
6338 ;ISSUE NOP COMMAND TO PORT A
6339
6340 040314 012760 000001 000000      MOV      #1,RMCS1(RO)
6341
6342 ;:*****
6343 ;VERIFY THAT ATTENTION FOR PORT A CLEARED
6344
6345 040322 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6346 040326 016037 000012 001126      MOV      RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
6347 040334 012737 000012 001122      MOV      #RMDS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6348 040342 060037 001122      ADD      RO,$BADDR ;ADD RHI1 BASE ADDRESS
6349 040346 005037 001124      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
6350 040352 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6351 040360 042737 077777 001164      BIC      #1,CATA,$TMP0 ;SAVE SPECIFIED BITS
6352 040366 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
6353 040374 001414      BEQ      73$ ;BR IF OK
6354 040376 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
6355 040404 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
6356 040412 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6357 040420 104061      ERROR 61 ;TYPE MESSAGE 61
6358 040422 005137 001250      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6359 040426 000240      73$: NOP
6360
6361 ;:*****
6362
6363 ;RELEASE THE DRIVE FROM PORT A
6364
6365 040430 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
6366 040436 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6367 040444 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
6368
6369 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6370
6371 040452 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
6372 040456 012737 000012 001122      MOV      #RMDS1,$BADDR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
6373 040464 060037 001122      ADD      RO,$BADDR ;ADD THE I/O BASE ADDRESS
6374 040470 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6375 040476 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
6376 040504 016037 000012 001170      MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6377 040512 042737 024001 001170      BIC      #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
6378 040520 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6379 040526 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6380 040534 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
6381 040542 016037 000012 001172      MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```


6382	040550	042737	024001	001172		BIC	#PIP!WRL!OM,\$TMP3	;CLEAR DONT CARES
6383	040556	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
6384	040564	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6385	040572	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6386	040600	001006				BNE	75\$;BR IF NOT
6387	040602	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6388	040606	001045				BNE	77\$;BR IF NOT
6389	040610	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6390	040612	000137	041012			JMP	79\$;BYPASS THE REST OF THE CHECKS
6391	040616	013737	001170	001126	75\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6392	040624	013737	001226	001240		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6393	040632	113760	001226	000010		MOV	PORTB,RMCS2(RO)	;SELECT PORT B.
6394	040640	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
6395	040644	001414				BEQ	76\$;BR IF ZERO
6396	040646	013737	001224	001240		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6397	040654	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
6398	040662	113760	001224	000010		MOV	PORTA,RMCS2(RO)	;SELECT PORT A.
6399	040670	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
6400	040674	001012				BNE	77\$;BR IF NOT
6401	040676	012737	177777	001254	76\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
6402	040704	012760	000011	000000		MOV	#11,RMCS1(RO)	;CLEAR THE DRIVE
6403	040712	012760	000013	000000		MOV	#13,RMCS1(RO)	;RELEASE THE DRIVE
6404	040720	104026				ERROR	26	;TYPE ERROR MESSAGE 26
6405	040722	013737	001170	001126	77\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6406	040730	013737	001224	001240		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
6407	040736	042737	100000	001126		BIC	#ATA,\$BDDAT	;DON'T CHECK THE ATTN BIT
6408	040744	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
6409	040752	001401				BEQ	78\$;BR IF OK FROM PORT A.
6410	040754	104007				ERROR	7	;REPORT ERROR
6411	040756	013737	001172	001126	78\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6412	040764	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
6413	040772	042737	100000	001126		BIC	#ATA,\$BDDAT	;DON'T CHECK THE ATTN BIT
6414	041000	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
6415	041006	001401				BEQ	79\$;BR IF OK
6416	041010	104007				ERROR	7	;REPORT ERROR
6417	041012	000240			79\$:	NOP		
6418								
6419								
6420								
6421								
6422								
6423	041014	113760	001226	000010		MOV	PORTB,RMCS2(RO)	;SELECT PORT B
6424	041022	013737	001226	001240		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6425	041030	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
6426	041034	016037	000012	001126		MOV	RMDS1(RO),\$BDDAT	;GET CONTENTS OF RMDS1
6427	041042	012737	000012	001122		MOV	#RMDS1,\$BDDAT	;FORM REGISTER ADDRESS OF ERROR MESSAGE
6428	041050	060037	001122			ADD	RO,\$BDDAT	;ADD RH11 BASE ADDRESS
6429	041054	012737	100000	001124		MOV	#ATA,\$GDDAT	;WHAT REGISTER SHOULD BE
6430	041062	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
6431	041070	042737	077777	001164		BIC	#ICATA,\$TMP0	;SAVE SPECIFIED BITS
6432	041076	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
6433	041104	001414				BEQ	80\$;BR IF OK
6434	041106	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
6435	041114	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
6436	041122	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
6437	041130	104062				ERROR	62	;TYPE MESSAGE 62

::*****
;VERIFY THAT ATTENTION FOR PORT B IS STIL SET

6438 041132 005137 001250
6439 041136 000240
6440 041140 000004

COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
80\$: NOP
1\$: SCOPE

6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6458
6459
6460

```
*****
*TEST 31 RESET ATTENTION 'B' BY GO TEST
*
* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
* ATTENTION BITS ARE SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S
* INTO RMD5.
*
* C. ISSUE A NOP COMMAND.
*
* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE
* ATTENTION BIT FOR PORT 'B' IS RESET, AND THE
* ATTENTION BIT FOR PORT 'A' IS STILL SET.
*****
```

6462 041142
6463 041142 005737 001300
6464 041146 001406
6465 041150 100002
6466 041152 000137 002676
6467 041156 012737 177777 001300
6468 041164 112737 000031 001102
6469 041172 012737 041214 001106
6470 041200 012737 041214 001110
6471 041206 012737 000031 001176
6472 041214 012706 001100

```
*****
*ST31:
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 #31,$STNM ;TEST NUMBER
MOV #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST31: MOV #STACK,$SP ;LOAD THE STACK POINTER
*****
```

6473
6474
6475
6476
6477
6478 041220 113760 001224 000010
6479 041226 005760 000012
6480 041232 001775
6481 041234 012760 177777 000014
6482 041242 005060 000014
6483 041246 013760 001226 000010
6484 041254 005760 000012
6485 041260 001775
6486 041262 012760 177777 000014
6487 041270 005060 000014
6488 041274 113760 001224 000010
6489 041302 005760 000012
6490 041306 001775
6491
6492
6493

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

E10

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 121
T31 RESET ATTENTION 'B' BY GO TEST

SEQ 0121

6494									
6495	041310	113760	001226	000010	MOV	PORTB, RMCS2(RO)	: SELECT PORT B		
6496	041316	013737	001226	001240	MOV	PORTB, PTNBR	: MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT		
6497	041324	005037	001250		CLR	CKERR	: CLEAR THE 'CHECK ERROR' INDICATOR		
6498	041330	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	: GET CONTENTS OF RMDS1		
6499	041336	012737	000012	001122	MOV	#RMDS1, \$BDAOR	: FORM REGISTER ADDRESS OF ERROR MESSAGE		
6500	041344	060037	001122		ADD	RO, \$BDAOR	: ADD RH11 BASE ADDRESS		
6501	041350	012737	100000	001124	MOV	#ATA, \$GDDAT	: WHAT REGISTER SHOULD BE		
6502	041356	013737	001126	001164	MOV	\$BDDAT, \$TMP0	: MOVE REGISTER CONTENTS TO '\$TMP0'		
6503	041364	042737	077777	001164	BIC	#ICATA, \$TMP0	: SAVE SPECIFIED BITS		
6504	041372	023737	001124	001164	CMP	\$GDDAT, \$TMP0	: COMPARE THE BITS		
6505	041400	001414			BEQ	67\$: BR IF OK		
6506	041402	013737	001126	001174	MOV	\$BDDAT, \$TMP4	: COPY 'BAD DATA'		
6507	041410	042737	100000	001174	BIC	#ATA, \$TMP4	: CLEAR THE MASKED BITS		
6508	041416	053737	0C1174	001124	BIS	\$TMP4, \$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT		
6509	041424	104010			ERROR	10	: REPORT THE ERROR		
6510	041426	005137	001250		COM	CKERR	: SET THE REGISTER COMPARE ERROR INDICATOR		
6511	041432	000240			67\$:	NO			
6512	041434	005737	001250		TST	CKERR	: WAS ATTENTION SET FOR B??		
6513	041440	001402			BEQ	:+6	: YES!!		
6514	041442	000137	042634		JMP	1\$: NO - BYPASS REST OF TEST		
6515	041446	113760	001224	000010	MOV	PORTA, RMCS2(RO)	: SELECT PORT A		
6516	041454	013737	001224	001240	MOV	PORTA, PTNBR	: MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT		
6517	041462	005037	001250		CLR	CKERR	: CLEAR THE 'CHECK ERROR' INDICATOR		
6518	041466	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	: GET CONTENTS OF RMDS1		
6519	041474	012737	000012	001122	MOV	#RMDS1, \$BDAOR	: FORM REGISTER ADDRESS OF ERROR MESSAGE		
6520	041502	060037	001122		ADD	RO, \$BDAOR	: ADD RH11 BASE ADDRESS		
6521	041506	012737	100000	001124	MOV	#ATA, \$GDDAT	: WHAT REGISTER SHOULD BE		
6522	041514	013737	001126	001164	MOV	\$BDDAT, \$TMP0	: MOVE REGISTER CONTENTS TO '\$TMP0'		
6523	041522	042737	077777	001164	BIC	#ICATA, \$TMP0	: SAVE SPECIFIED BITS		
6524	041530	023737	001124	001164	CMP	\$GDDAT, \$TMP0	: COMPARE THE BITS		
6525	041536	001414			BEQ	69\$: BR IF OK		
6526	041540	013737	001126	001174	MOV	\$BDDAT, \$TMP4	: COPY 'BAD DATA'		
6527	041546	042737	100000	001174	BIC	#ATA, \$TMP4	: CLEAR THE MASKED BITS		
6528	041554	053737	001174	001124	BIS	\$TMP4, \$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT		
6529	041562	104010			ERROR	10	: REPORT THE ERROR		
6530	041564	005137	001250		COM	CKERR	: SET THE REGISTER COMPARE ERROR INDICATOR		
6531	041570	000240			69\$:	NO			
6532	041572	005737	001250		TST	CKERR	: WAS ATTENTION SET FOR A??		
6533	041576	001402			BEQ	:+6	: YES!!		
6534	041600	000137	042634		JMP	1\$: NO - BYPASS REST OF TEST		
6535									
6536									
6537									
6538									
6539									
6540	041604	113760	001226	000010	MOV	PORTB, RMCS2(RO)	: SELECT PORT B		
6541	041612	013737	001226	001242	MOV	PORTB, SEIZPT	: STORE SEIZING PORT'S ADDRESS		
6542	041620	005060	000012		CLR	RMDS1(RO)	: WRITE RMDS1		
6543	041624	113760	001224	000010	MOV	PORTA, RMCS2(RO)	: SELECT PORT A		
6544	041632	013737	001224	001240	MOV	PORTA, PTNBR	: MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT		
6545	041640	013737	001224	001244	MOV	PORTA, OPPRT	: 'OPPOSITE' PORT ADDRESS		
6546	041646	016037	000012	001126	MOV	RMDS1(RO), \$BDDAT	: SEE IF DRIVE SEIZED BY PORT B		
6547	041654	010037	001122		MOV	RO, \$BDAOR	: RH11 BASE ADDRESS		
6548	041660	062737	000012	001122	ADD	#RMDS1, \$BDAOR	: GENERATE BAD REGISTER ADDRESS		
6549	041666	005037	001124		CLR	\$GDDAT	: REGISTER SHOULD BE ZERO		

::*****

; SEIZE THE DRIVE THROUGH PORT B

F10

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 122
T31 RESET ATTENTION 'B' BY GO TEST

SEQ 0122

```

6550 041672 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;IS THE REGISTER ZERO
6551 041700 001403                    BEQ      71$                ;BR IF IT IS
6552 041702 104004                    ERROR   4                    ;REPORT THE ERROR
6553 041704 000137 042634              JMP      1$                  ;BYPASS REST OF THE SUBTEST
6554 041710                                71$:
6555 041710 113760 001226 000010      MOV     PORTB, RMCS2(R0)    ;SELECT PORT B
6556 041716 013737 001226 001240      MOV     PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6557 041724 016037 000012 001126      MOV     RMDS1(R0), $BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
6558 041732 042737 020001 001126      BIC     #OM:PIP,$BDDAT     ;CLEAR DONT CARE BITS
6559 041740 012737 011700 001124      MOV     #MOL:PGM:DPR:DRY:VV,$GDDAT ;EXPECTED STATUS
6560 041746 013737 001124 001166      MOV     $GDDAT,$TMP1      ;USE GOOD DATA AS A MASK
6561 041754 005137 001166              COM     $TMP1              ;COMPLEMENT THE EXPECTED STATUS
6562 041760 013737 001126 001164      MOV     $BDDAT,$TMP0      ;SAVE THE ACTUAL STATUS
6563 041766 043737 001166 001164      BIC     $TMP1,$TMP0       ;CLEAR UNWANTED BITS
6564 041774 023737 001124 001164      CMP     $GDDAT,$TMP0      ;ARE THE EXPECTED STATUS BITS SET ?
6565 042002 001401                    BEQ     72$                ;BR IF THEY ARE
6566 042004 104005                    ERROR   5                    ;REPORT THE ERROR
6567 042006 000240                                72$:
6568                                NOP
6569                                ;:*****
6570                                ;ISSUE NOP COMMAND TO PORT B
6571                                ;
6572                                MOV     #1,RMCS1(R0)
6573 042010 012760 000001 000000
6574                                ;:*****
6575                                ;VERIFY THAT ATTENTION FOR PORT B CLEARED
6576                                ;
6577                                ;
6578 042016 005037 001250              CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
6579 042022 016037 000012 001126      MOV     RMDS1(R0), $BDDAT  ;GET CONTENTS OF RMDS1
6580 042030 012737 000012 001122      MOV     #RMDS1,$BDDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6581 042036 060037 001122              ADD     R0,$BDDADR        ;ADD R#11 BASE ADDRESS
6582 042042 005037 001124              CLR     $GDDAT            ;WHAT REGISTER SHOULD BE
6583 042046 013737 001126 001164      MOV     $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
6584 042054 042737 077777 001164      BIC     #ICATA,$TMP0      ;SAVE SPECIFIED BITS
6585 042062 023737 001124 001164      CMP     $GDDAT,$TMP0      ;COMPARE THE BITS
6586 042070 001414                    BEQ     73$                ;BR IF OK
6587 042072 013737 001126 001174      MOV     $BDDAT,$TMP4      ;COPY 'BAD DATA'
6588 042100 042737 100000 001174      BIC     #ATA,$TMP4        ;CLEAR THE MASKED BITS
6589 042106 053737 001174 001124      BIS     $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
6590 042114 104061                    ERROR   61                   ;TYPE MESSAGE 61
6591 042116 005137 001250              COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
6592 042122 000240                                73$:
6593                                NOP
6594                                ;:*****
6595                                ;
6596                                ;RELEASE THE DRIVE FROM PORT B
6597                                ;
6598 042124 113760 001226 000010      MOV     PORTB, RMCS2(R0)    ;SELECT PORT B
6599 042132 013737 001226 001240      MOV     PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6600 042140 012760 000013 000000      MOV     #13,RMCS1(R0)     ;ISSUE RELEASE THROUGH PORT B
6601                                ;
6602                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6603                                ;
6604 042146 005037 001254              CLR     RELERR            ;CLEAR THE 'RELEASE ERROR' INDICATOR
6605 042152 012737 000012 001122      MOV     #RMDS1,$BDDADR    ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT

```

G10

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 123
T31 RESET ATTENTION 'B' BY GO TEST

SEG 0123

```

6606 042160 060037 001122      ADD      RO,$B0ADR      ;ADD THE I/O BASE ADDRESS
6607 042164 012737 011700 001124  MOV      #MOL:PGM:OPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
6608 042172 113760 001224 000010  MOVVB   PORTA,AMCS2(RO) ;SELECT PORT A.
6609 042200 016037 000012 001170  MOV      RMDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6610 042206 042737 024001 001170  BIC     #PIP:WAL:OM,$TMP2 ;CLEAR DONT CARES
6611 042214 013737 001170 001164  MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6612 042222 042737 100100 001164  BIC     #ATA:VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6613 042230 113760 001226 000010  MOVVB   PORTB,AMCS2(RO) ;SELECT PORT B.
6614 042236 016037 000012 001172  MOV      RMDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6615 042244 042737 024001 001172  BIC     #PIP:WAL:OM,$TMP3 ;CLEAR DONT CARES
6616 042252 013737 001172 001166  MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
6617 042260 042737 100100 001166  BIC     #ATA:VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6618 042266 023737 001164 001166  CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6619 042274 001006 75$      BNE     75$ ;BR IF NOT
6620 042276 005737 001164 75$      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6621 042302 001045 77$      BNE     77$ ;BR IF NOT
6622 042304 104046 76$      ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6623 042306 000137 042506 79$      JMP     79$ ;BYPASS THE REST OF THE CHECKS
6624 042312 013737 001170 001126 75$:    MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6625 042320 013737 001226 001240 75$:    MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6626 042326 113760 001226 000010 75$:    MOVVB  PORTB,AMCS2(RO) ;SELECT PORT B.
6627 042334 005737 001164 75$:    TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6628 042340 001414 76$      BEQ     76$ ;BR IF ZERO
6629 042342 013737 001224 001240 75$:    MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6630 042350 013737 001172 001126 75$:    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6631 042356 113760 001224 000010 75$:    MOVVB  PORTA,AMCS2(RO) ;SELECT PORT A.
6632 042364 005737 001166 75$:    TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6633 042370 001012 77$      BNE     77$ ;BR IF NOT
6634 042372 012737 177777 001254 76$:    MOV     #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
6635 042400 012760 000011 000000 76$:    MOV     #11,AMCS1(RO) ;CLEAR THE DRIVE
6636 042406 012760 000013 000000 76$:    MOV     #13,AMCS1(RO) ;RELEASE THE DRIVE
6637 042414 104026 76$      ERROR   26 ;TYPE ERROR MESSAGE 26
6638 042416 013737 001170 001126 77$:    MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6639 042424 013737 001224 001240 77$:    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
6640 042432 042737 100000 001126 77$:    BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6641 042440 023737 001124 001126 77$:    CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
6642 042446 001401 78$      BEQ     78$ ;BR IF OK FROM PORT A.
6643 042450 104007 77$      ERROR   7 ;REPORT ERROR
6644 042452 013737 001172 001126 78$:    MOV     $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6645 042460 013737 001226 001240 78$:    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
6646 042466 042737 100000 001126 78$:    BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
6647 042474 023737 001124 001126 78$:    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
6648 042502 001401 79$      BEQ     79$ ;BR IF OK
6649 042504 104007 79$      ERROR   7 ;REPORT ERROR
6650 042506 000240 79$:    NOP
6651
6652 ;*****
6653
6654 ;VERIFY THAT ATTENTION FOR PORT A IS STIL SET
6655
6656 042510 113760 001224 000010 79$:    MOVVB  PORTA,AMCS2(RO) ;SELECT PORT A
6657 042516 013737 001224 001240 79$:    MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6658 042524 005037 001250 79$:    CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6659 042530 016037 000012 001126 79$:    MOV     RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
6660 042536 012737 000012 001126 79$:    MOV     #RMDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6661 042544 060037 001122 79$:    ADD     RO,$B0ADR ;ADD RHI1 BASE ADDRESS

```

H10

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 124
T31 RESET ATTENTION 'B' BY GO TEST

SEQ 0124

```

6662 042550 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
6663 042556 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6664 042564 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
6665 042572 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6666 042600 001414 BEQ B0$ ;BR IF OK
6667 042602 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6668 042610 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6669 042616 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6670 042624 104062 ERROR 62 ;TYPE MESSAGE 62
6671 042626 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6672 042632 000240 B0$: NOP
6673 042634 000004 1$: SCOPE

```

```

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

```

```

6689 042636 005737 001300 †ST32:
6690 042636 001406 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6691 042642 100002 BEQ 2$ ;BR IF NOT
6692 042644 000137 002676 BPL 1$ ;BR IF JUST ENTERED TEST
6693 042646 012737 177777 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6694 042652 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6695 042660 112737 000032 001102 2$: MOV #32,$STNM ;TEST NUMBER
6696 042666 012737 042710 001106 MOV #TEST32,$LPADR ;LOAD LOOP ON TEST ADDRESS
6697 042674 012737 042710 001110 MOV #TEST32,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6698 042702 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
6699 042710 012706 TEST32: MOV #STACK,SP ;LOAD THE STACK POINTER
6700 *****
6701 ;SET ATTENTION BITS FOR BOTH PORTS
6702
6703
6704 042714 113760 001224 000010 66$: MOV# PORTA, RMCS2(RO) ;SELECT PORT 64$
6705 042722 005760 000012 TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
6706 042726 001775 BEQ 66$
6707 042730 012760 177777 000014 MOV #-1, RMER1(RO) ;FORCE ERRORS
6708 042736 005060 000014 CLR RMEB1(RO) ;CLEAR THE ERRORS
6709 042742 013760 001226 000010 64$: MOV PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
6710 042750 005760 000012 TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6711 042754 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
6712 042756 012760 177777 000014 MOV #-1, RMER1(RO) ;FORCE ERRORS ON PORT 65$
6713 042764 005060 000014 CLR RMEB1(RO) ;CLEAR THE ERRORS
6714 042770 113760 001224 000010 65$: MOV# PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
6715 042776 005760 000012 TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
6716 043002 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT

```

```

6718
6719
6720
6721 043004 113760 001224 000010
6722 043012 013737 001224 001240
6723 043020 005037 001250
6724 043024 016037 000012 001126
6725 043032 012737 000012 001122
6726 043040 060037 001122
6727 043044 012737 100000 001124
6728 043052 013737 001126 001164
6729 043060 042737 077777 001164
6730 043066 023737 001124 001164
6731 043074 001414
6732 043076 013737 001126 001174
6733 043104 042737 100000 001174
6734 043112 053737 001174 001124
6735 043120 104010
6736 043122 005137 001250
6737 043126 000240
6738 043130 005737 001250
6739 043134 001402
6740 043136 000137 044112
6741 043142 113760 001226 000010
6742 043150 013737 001226 001240
6743 043156 005037 001250
6744 043162 016037 000012 001126
6745 043170 012737 000012 001122
6746 043176 060037 001122
6747 043202 012737 100000 001124
6748 043210 013737 001126 001164
6749 043216 042737 077777 001164
6750 043224 023737 001124 001164
6751 043232 001414
6752 043234 013737 001126 001174
6753 043242 042737 100000 001174
6754 043250 053737 001174 001124
6755 043256 104010
6756 043260 005137 001250
6757 043264 000240
6758 043266 005737 001250
6759 043272 001402
6760 043274 000137 044112
6761
6762
6763
6764
6765
6766 043300 005037 001254
6767 043304 012737 000012 001122
6768 043312 060037 001122
6769 043316 012737 111700 001124
6770 043324 113760 001224 000010
6771 043332 016037 000012 001170
6772 043340 042737 024001 001170
6773 043346 013737 001170 001164

```

```

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

```

```

MOV B PORTA, RMCS2(R0) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV #RMS1, $BDDAT ;GET CONTENTS OF RMS1
MOV #RMS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDAOR ;ADD RHI1 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 67$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

67$:
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 B PORTB, RMCS2(R0) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV #RMS1, $BDDAT ;GET CONTENTS OF RMS1
MOV #RMS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDAOR ;ADD RHI1 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, $TMP0 ;COMPARE THE BITS
BEQ 69$ ;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

69$:
NOP
TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
BEQ .+6 ;BR IF IT WAS
JMP IS ;BYPASS REST OF TEST IF NOT

```

```

;*****

```

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

```

CLR RELEA ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RMS1, $BDAOR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
ADD R0, $BDAOR ;ADD THE I/O BASE ADDRESS
MOV #11700, $GDDAT ;COMPARISON CONSTANT
MOV B PORTA, RMCS2(R0) ;SELECT PORT A
MOV RMS1(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'

```

J10

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046)
T32

21-NOV-77 13:53 PAGE 126
TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT

SEQ 0126

6774	043354	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6775	043362	113760	001226	000010		MOVB	PORTB,\$RMC52(RO)	;SELECT PORT B.
6776	043370	016037	000012	001172		MOV	RMDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
6777	043376	042737	024001	001172		BIC	#PIP!WRL!OM,\$TMP3	;CLEAR DONT CARES
6778	043404	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
6779	043412	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
6780	043420	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6781	043426	001006				BNE	71\$;BR IF NOT
6782	043430	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6783	043434	001045				BNE	73\$;BR IF NOT
6784	043436	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6785	043440	000137	043624			JMP	75\$;BYPASS THE REST OF THE CHECKS
6786	043444	013737	001170	001126	71\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6787	043452	013737	001226	001240		MOV	PORTB,\$PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6788	043460	113760	001226	000010		MOVB	PORTB,\$RMC52(RO)	;SELECT PORT B.
6789	043466	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
6790	043472	001414				BEQ	72\$;BR IF ZERO
6791	043474	013737	001224	001240		MOV	PORTA,\$PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6792	043502	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE 0LT
6793	043510	113760	001224	000010		MOVB	PORTA,\$RMC52(RO)	;SELECT PORT A.
6794	043516	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
6795	043522	001012				BNE	73\$;BR IF NOT
6796	043524	012737	177777	001254	72\$:	MOV	#-1,\$RELERR	;SET 'RELEASE ERROR' INDICATOR
6797	043532	012760	000011	000000		MOV	#11,\$RMC51(RO)	;CLEAR THE DRIVE
6798	043540	012760	000013	000000		MOV	#13,\$RMC51(RO)	;RELEASE THE DRIVE
6799	043546	104026				ERROR	26	;TYPE ERROR MESSAGE 26
6800	043550	013737	001170	001126	73\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMDS1 READ
6801	043556	013737	001224	001240		MOV	PORTA,\$PTNBR	;CHANGE PORT NUMBER
6802	043564	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
6803	043572	001401				BEQ	74\$;BR IF OK FROM PORT A.
6804	043574	104007				ERROR	7	;REPORT ERROR
6805	043576	013737	001172	001126	74\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
6806	043604	013737	001226	001240		MOV	PORTB,\$PTNBR	;CHANGE PORT NUMBER
6807	043612	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
6808	043620	001401				BEQ	75\$;BR IF OK
6809	043622	104007				ERROR	7	;REPORT ERROR
6810	043624	000240			75\$:	NOB		
6811	043626	005737	001254			TST	\$RELERR	;WAS DRIVE IN NEUTRAL ?
6812	043632	001402				BEQ	76\$;BR IF IT WAS
6813	043634	000137	044112			JMP	1\$;BYPASS RESET OF TEST
6814						;*****		
6815						;ISSUE THE MASSBUS INIT		
6816								
6817	043640	012760	000040	000010		MOV	#CLR,\$RMC52(RO)	;ISSUE A MASSBUS INIT
6818								
6819						;*****		
6820						;CHECK E ATTENTION BITS OF BOTH PORTS		
6821								
6822	043646	113760	001224	000010		MOVB	PORTA,\$RMC52(RO)	;SELECT PORT A
6823	043654	013737	001224	001240		MOV	PORTA,\$PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6824	043662	005037	001250			CLR	\$CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
6825	043666	016037	000012	001126		MOV	RMDS1(RO),\$BDDAT	;GET CONTENTS OF RMDS1
6826	043674	012737	000012	001122		MOV	#RMDS1,\$BDDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
6827	043702	060037	001122			ADD	RO,\$BDDADR	;ADD RM11 BASE ADDRESS
6828	043706	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
6829	043712	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'


```

6830 043720 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
6831 043726 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6832 043734 001414 BEQ 76$ ;BR IF OK
6833 043736 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6834 043744 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6835 043752 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6836 043760 104051 ERROR 51 ;TYPE MESSAGE 51
6837 043762 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6838 043766 000240 76$: NOP
6839 043770 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
6840 043776 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6841 044004 005037 001250 CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6842 044010 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
6843 044016 012737 000012 001122 MOV #RMDS1, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6844 044024 060037 001122 ADD R0, $BDDADR ;ADD RHI1 BASE ADDRESS
6845 044030 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
6846 044034 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6847 044042 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
6848 044050 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6849 044056 001414 BEQ 78$ ;BR IF OK
6850 044060 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6851 044066 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6852 044074 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6853 044102 104051 ERROR 51 ;TYPE MESSAGE 51
6854 044104 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6855 044110 000240 79$: NOP
6856 044112 000004 1$: SCOPE ;LOOP ?

```

```

6860 *****
6861 *TEST 33 RESET ATTENTION 'A' & 'B' BY RMAS
6862 *
6863 *VERIFY THAT BOTH ATTENTION BITS CAN BE RESET BY WRITING THE
6864 *APPROPRIATE BIT IN THE ATTENTION SUMMARY REGISTER.
6865 *
6866 * A. SET THE ATTENTION BITS FOR BOTH PORTS.
6867 *
6868 * B. VERIFY THE DRIVE IS IN NEUTRAL.
6869 *
6870 * C. WRITE THE DRIVE'S ATTENTION BIT IN RMAS. VERIFY
6871 * THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMAS.
6872 *
6873 *****
6874 †ST33:
6875 044114 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6876 044120 001406 BEQ 2$ ;BR IF NOT
6877 044122 100002 BPL 1$ ;BR IF JUST ENTERED TEST
6878 044124 000137 002676 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6879 044130 012737 177777 001300 1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
6880 044136 112737 000033 001102 2$: MOVB #33, $STNM ;TEST NUMBER
6881 044144 012737 044166 001106 MOV #TEST33, $LPADR ;LOAD LOOP ON TEST ADDRESS
6882 044152 012737 044166 001110 MOV #TEST33, $LPERR ;LOAD LOOP ON ERROR ADDRESS
6883 044160 012737 000002 001176 MOV #2, $TIMES ;DO 2. ITERATIONS
6884 044166 012706 001100 TEST33: MOV #STACK, SP ;LOAD THE STACK POINTER
6885 ;*****

```



```

6886
6887
6888
6889 044172 113760 001224 000010
6890 044200 005760 000012
6891 044204 001775
6892 044206 012760 177777 000014
6893 044214 005060 000014
6894 044220 013760 001226 000010
6895 044226 005760 000012
6896 044232 001775
6897 044234 012760 177777 000014
6898 044242 005060 000014
6899 044246 113760 001224 000010
6900 044254 005760 000012
6901 044260 001775
6902
6903
6904 044262 113760 001224 000010
6905 044270 013737 001224 001240
6906 044276 005037 001250
6907 044302 016037 000012 001126
6908 044310 012737 000012 001122
6909 044316 060037 001122
6910 044322 012737 100000 001124
6911 044330 013737 001126 001164
6912 044336 042737 077777 001164
6913 044344 023737 001124 001164
6914 044352 001414
6915 044354 013737 001126 001174
6916 044362 042737 100000 001174
6917 044370 053737 001174 001124
6918 044376 104010
6919 044400 005137 001250
6920 044404 000240
6921 044406 005737 001250
6922 044412 001402
6923 044414 000137 045236
6924 044420 113760 001226 000010
6925 044426 013737 001226 001240
6926 044434 005037 001250
6927 044440 016037 000012 001126
6928 044446 012737 000012 001122
6929 044454 060037 001122
6930 044460 012737 100000 001124
6931 044466 013737 001126 001164
6932 044474 042737 077777 001164
6933 044502 023737 001124 001164
6934 044510 001414
6935 044512 013737 001126 001174
6936 044520 042737 100000 001174
6937 044526 053737 001174 001124
6938 044534 104010
6939 044536 005137 001250
6940 044542 000240
6941 044544 005737 001250

;SET ATTENTION BITS FOR BOTH PORTS
66$: MOVB PORTA, RMCS2(RO) ;SELECT PORT 64$
TST RMD51(RO) ;MAKE SURE DRIVE AVAILABLE
BEQ 66$
MOV #-1, RMR1(RO) ;FORCE ERRORS
CLR RMR1(RO) ;CLEAR THE ERRORS
64$: MOV PORTB, RMCS2(RO) ;SELECT THE OTHER PORT
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1, RMR1(RO) ;FORCE ERRORS ON PORT 65$
CLR RMR1(RO) ;CLEAR THE ERRORS
65$: MOVB PORTA, RMCS2(RO) ;SELECT PORT "64$" AGAIN
TST RMD51(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
MOV #RMD51, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAADR ;ADD RHI1 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 67$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
67$: NOP
TST CKERR ;WAS ATA SET FOR A??
BEQ .+6 ;YES - CONTINUE
JMP 1$ ;BYPASS REST OF TEST
MOVB PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
MOV #RMD51, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAADR ;ADD RHI1 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, $TMP0 ;COMPARE THE BITS
BEQ 69$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
69$: NOP
TST CKERR ;WAS ATA SET FOR B??

```



```

6998
6999 045116 013760 001236 000016      MOV      ASR1,RMAS(RO)
7000
7001      ;VERIFY THAT BOTH ATTENTIONS ARE RESET BY READING RMAS
7002 045124 016037 000016 001126      MOV      RMAS(RO),%BDDAT      ;GET ATTENTION SUMMARY
7003 045132 033737 001236 001126      BIT      ASR1,%BDDAT         ;IS THE ATTENTION RESET ??
7004 045140 001414                BEQ      2$                  ;YES !!
7005 045142 010037 001122                MOV      RO,%BDADR          ;SETUP REGISTER ADDRESS
7006 045146 062737 000016 001122      ADD      #RMAS,%BDADR
7007 045154 013737 001126 001124      MOV      %BDDAT,%GDDAT      ;SETUP EXPECTED DATA
7008 045162 043737 001236 001124      BIC      ASR1,%GDDAT        ;RESET THIS DRIVES BIT
7009 045170 104060                ERROR   60                  ;ATTENTION NOT CLEARED BY RMAS
7010
7011 045172      2$:
7012      ;WAIT FOR THE DRIVES TO RELEASE BY TIMEOUT
7013
7014 045172 113760 001224 000010      MOV      PORTA,RMCS2(RO)    ;SELECT PORT A
7015 045200 013737 001224 001240      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7016 045206 005760 000012      JS:      TST      RMDS1(RO) ;MAKE SURE DEVICE IS AVAILABLE
7017 045212 001775                BEQ      3$
7018 045214 113760 001226 000010      MOV      PORTB,RMCS2(RO)    ;SELECT PORT B
7019 045222 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7020 045230 005760 000012      JS:      TST      RMDS1(RO) ;MAKE SURE DEVICE IS AVAILABLE
7021 045234 001775                BEQ      4$
7022 045236 000004      4$:
7023      SCOPE
7024
7025      ;*****
7026      *TEST 34      PORT 'A' ALTERNATE ATTENTION PATH TEST
7027      *
7028      ;VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
7029      *
7030      * A. SET THE ATTENTION BIT FOR PORT 'A'.
7031      *
7032      * B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
7033      *
7034      * C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
7035      * FOR THE DRIVE IS SET.
7036      *
7037      ;*****
7038      *ST34:
7039 045240 005737 001300      TST      KYBCTL             ;PERFORMING ONLY SINGLE TESTS ?
7040 045244 001406                BEQ      2$                  ;BR IF NOT
7041 045246 100002                BPL      1$                  ;BR IF JUST ENTERED TEST
7042 045250 000137 002676      JMP      EXEC               ;RETURN & GET NEXT TEST NUMBER
7043 045254 012737 177777 001300      1$:      MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
7044 045262 112737 000034 001102      2$:      MOV      #34,%STSTNM     ;TEST NUMBER
7045 045270 012737 045312 001106      MOV      #TEST34,%SLPADR    ;LOAD LOOP ON TEST ADDRESS
7046 045276 012737 045312 001110      MOV      #TEST34,%SLPERR    ;LOAD LOOP ON ERROR ADDRESS
7047 045304 012737 000031 001176      MOV      #25,%STIMES        ;DO 25 ITERATIONS
7048 045312 012706 001100      TEST34: MOV      #STACK,SP        ;LOAD THE STACK POINTER
7049
7050      ;CLEAR ATTENTION BITS FOR BOTH PORTS
7051
7052 045316 113760 001224 000010      MOV      PORTA,RMCS2(RO)    ;SELECT PORT #A
7053 045324 005060 000012      CLR      RMCS1(RO)          ;SEIZE THE DRIVE

```

7054	045330	012760	000011	000000	MOV	#11,RMCS1(RO)	:ISSUE DRIVE CLEAR
7055	045336	012760	000013	000000	MOV	#13,RMCS1(RO)	:RELEASE THE DRIVE
7056	045344	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	:SELECT PORT #B
7057	045352	005060	000012		CLR	RMDS1(RO)	:SEIZE THE DRIVE THROUGH PORT 'B'
7058	045356	012760	000011	000000	MOV	#11,RMCS1(RO)	:ISSUE DRIVE CLEAR
7059	045364	012760	000013	000000	MOV	#13,RMCS1(RO)	:RELEASE THE DRIVE
7060	045372	113760	001224	000010	MOVB	PORTA,RMCS2(RO)	:SELECT PORT A
7061	045400	012760	177777	000014	MOV	#-1,RMER1(RO)	:SET ERRORS TO FORCE ATTN BIT ON PORT A
7062	045406	005060	000014		CLR	RMER1(RO)	:CLEAR THE ERRORS
7063	045412	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	:SELECT PORT B
7064	045420	005760	000012		15: TST	RMDS1(RO)	:WAIT FOR DRIVE TO RETURN TO NEUTRAL
7065	045424	001775			BEQ	15	:BR IF STILL SEIZED BY PORT A
7066	045426	012737	000016	001122	MOV	#RMAS,\$B0ADR	:FORM ADDRESS OF ATTN REG IF ERROR
7067	045434	060037	001122		ADD	RO,\$B0ADR	:ADD THE ADDRESS BASE
7068	045440	013737	001236	001124	MOV	ASR1,\$GDDAT	:GOOD DATA FOR ERROR MESSAGE
7069	045446	013737	001236	001166	MOV	ASR1,\$TMP1	:MAKE DATA COMPARE MASK
7070	045454	005137	001166		COM	\$TMP1	:COMPLEMENT IT
7071	045460	012737	045514	001110	MOV	#25,\$LPERR	:LOAD LOOP ON ERROR ADDRESS
7072	045466	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	:SELECT PORT B
7073	045474	013737	001226	001240	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7074	045502	013737	001226	001242	MOV	PORTB,SEIZPT	: 'SEIZED' PORT ADDRESS
7075	045510	005060	000012		CLR	RMDS1(RO)	:SEIZE THE DRIVE THROUGH PORT B
7076	045514	016037	000016	001126	25: MOV	RMAS(RO),\$B0DAT	:GET THE CONTENTS OF THE ATTENTION REG
7077	045522	013737	001126	001164	MOV	\$B0DAT,\$TMP0	:PUT CONTENTS INTO WORKING LOCATION
7078	045530	043737	001166	001164	BIC	\$TMP1,\$TMP0	:CLEAR OTHER BITS
7079	045536	023737	001124	001164	CMP	\$GDDAT,\$TMP0	:SEE IF ATTN BIT FOR DRIVE SET
7080	045544	001401			BEQ	35	:BR IF SET
7081	045546	104053			ERROR	53	:REPORT THE ERROR
7082	045550				35:		
7083							
7084							:RELEASE THE DRIVE FROM PORT B
7085							
7086	045550	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	:SELECT PORT B
7087	045556	013737	001226	001240	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7088	045564	012760	000013	000000	MOV	#13,RMCS1(RO)	:ISSUE RELEASE THROUGH PORT B
7089							
7090							:VERIFY THAT THE DRIVE IS IN NEUTRAL
7091							
7092	045572	005037	001254		CLR	RELERR	:CLEAR THE 'RELEASE ERROR' INDICATOR
7093	045576	012737	000012	001122	MOV	#RMDS1,\$B0ADR	:FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7094	045604	060037	001122		ADD	RO,\$B0ADR	:ADD THE I/O BASE ADDRESS
7095	045610	012737	011700	001124	MOV	#M0L!PGM!DPR!DRY!VV,\$GDDAT	:COMPARISON CONSTANT
7096	045616	113760	001224	000010	MOVB	PORTA,RMCS2(RO)	:SELECT PORT A.
7097	045624	016037	000012	001170	MOV	RMDS1(RO),\$TMP2	:GET THE DRIVE STATUS REGISTER FROM PORT A.
7098	045632	042737	024001	001170	BIC	#PIP!WRL!OM,\$TMP2	:CLEAR DONT CARES
7099	045640	013737	001170	001164	MOV	\$TMP2,\$TMP0	:COPY IT INTO '\$TMP0'
7100	045646	042737	100100	001164	BIC	#ATA!VV,\$TMP0	:CLEAR PORT DEPENDENT BITS FROM THE COPY
7101	045654	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	:SELECT PORT B.
7102	045662	016037	000012	001172	MOV	RMDS1(RO),\$TMP3	:GET THE DRIVE STATUS REGISTER FROM PORT B.
7103	045670	042737	024001	001172	BIC	#PIP!WRL!OM,\$TMP3	:CLEAR DONT CARES
7104	045676	013737	001172	001166	MOV	\$TMP3,\$TMP1	:COPY IT INTO '\$TMP1'
7105	045704	042737	100100	001166	BIC	#ATA!VV,\$TMP1	:CLEAR PORT DEPENDENT BITS FROM THE COPY
7106	045712	023737	001164	001166	CMP	\$TMP0,\$TMP1	:IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7107	045720	001006			BNE	64\$:BR IF NOT
7108	045722	005737	001164		ST	\$TMP0	:REGISTERS ARE THE SAME: ARE THEY ZERO ?
7109	045726	001045			BNE	66\$:BR IF NOT

7110	045730	104046				ERROR	46	:REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7111	045732	000137	046132			JMP	68\$:BYPASS THE REST OF THE CHECKS
7112	045736	013737	001170	001126	64\$:	MOV	\$TMP2,\$BDDAT	:SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7113	045744	013737	001226	001240		MOV	PORTB,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7114	045752	113760	001226	000010		MOVB	PORTB,RMCS2(RO)	:SELECT PORT B.
7115	045760	005737	001164			TST	\$TMP0	:SEE IF STATUS EQ 0 FROM PORT A.
7116	045764	001414				BEQ	65\$:BR IF ZERO
7117	045766	013737	001224	001240		MOV	PORTA,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7118	045774	013737	001172	001126		MOV	\$TMP3,\$BDDAT	: 'BAD DATA' FOR ERROR TYPE OUT
7119	046002	113760	001224	000010		MOVB	PORTA,RMCS2(RO)	:SELECT PORT A.
7120	046010	005737	001166			TST	\$TMP1	:SEE IF STATUS EQ ZERO FROM PORT B.
7121	046014	001012				BNE	66\$:BR IF NOT
7122	046016	012737	177777	001254	65\$:	MOV	#-1,RELEA	:SET 'RELEASE ERROR' INDICATOR
7123	046024	012760	000011	000000		MOV	#11,RMCS1(RO)	:CLEAR THE DRIVE
7124	046032	012760	000013	000000		MOV	#13,RMCS1(RO)	:RELEASE THE DRIVE
7125	046040	104026				ERROR	26	:TYPE ERROR MESSAGE 26
7126	046042	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT	:LOOK FOR BIT FAILURES WHEN RMDS1 READ
7127	046050	013737	001224	001240		MOV	PORTA,PTNBR	:CHANGE PORT NUMBER
7128	046056	042737	100000	001126		BIC	#ATA,\$BDDAT	:DON'T CHECK THE ATTN BIT
7129	046064	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:ALL BITS OK ?
7130	046072	001401				BEQ	67\$:BR IF OK FROM PORT A.
7131	046074	104007				ERROR	7	:REPORT ERROR
7132	046076	013737	001172	001126	67\$:	MOV	\$TMP3,\$BDDAT	:CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7133	046104	013737	001226	001240		MOV	PORTB,PTNBR	:CHANGE PORT NUMBER
7134	046112	042737	100000	001126		BIC	#ATA,\$BDDAT	:DON'T CHECK THE ATTN BIT
7135	046120	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:SEE IF READ OK FROM PORT B.
7136	046126	001401				BEQ	68\$:BR IF OK
7137	046130	104007				ERROR	7	:REPORT ERROR
7138	046132	000240			68\$:	NOP		
7139	046134	000004				SCOPE		:LOOP ?

```

7140
7141
7142 :*****
7143 :*TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST
7144 :*
7145 :*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
7146 :*
7147 :* A. SET THE ATTENTION BIT FOR PORT 'B'.
7148 :*
7149 :* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
7150 :*
7151 :* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
7152 :* FOR THE DRIVE IS SET.
7153 :*
7154 :*****

```

7154	046136					TST35:		
7155	046136	005737	001300			TST	KYBCTL	:PERFORMING ONLY SINGLE TESTS ?
7156	046142	001406				BEQ	2\$:BR IF NOT
7157	046144	100002				BPL	1\$:BR IF JUST ENTERED TEST
7158	046146	000137	002676			JMP	EXEC	:RETURN & GET NEXT TEST NUMBER
7159	046152	012737	177777	001300	1\$:	MOV	#-1,KYBCTL	:SET SINGLE TEST INDICATOR
7160	046160	112737	000035	001102	2\$:	MOVB	#35,\$STSTM	:TEST NUMBER
7161	046166	012737	046210	001106		MOV	#TEST35,\$LPADR	:LOAD LOOP ON TEST ADDRESS
7162	046174	012737	046210	001110		MOV	#TEST35,\$LPERR	:LOAD LOOP ON ERROR ADDRESS
7163	046202	012737	000031	001176		MOV	#25,\$TIMES	:DO 25 ITERATIONS
7164	046210	012706	001100		TEST35:	MOV	#STACK,SP	:LOAD THE STACK POINTER
7165								

```

7166                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
7167
7168 046214 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ; SELECT PORT #A
7169 046222 005060 000012 000011      CLR    RMDS1(RO)       ; SEIZE THE DRIVE
7170 046226 012760 000011 000000      MOV    #11, RMCS1(RO)  ; ISSUE DRIVE CLEAR
7171 046234 012760 000013 000000      MOV    #13, RMCS1(RO)  ; RELEASE THE DRIVE
7172 046242 113760 001226 000010      MOVB   PORTB, RMCS2(RO); SELECT PORT #B
7173 046250 005060 000012 000000      CLR    RMDS1(RO)       ; SEIZE THE DRIVE THROUGH PORT 'B'
7174 046254 012760 000011 000000      MOV    #11, RMCS1(RO)  ; ISSUE DRIVE CLEAR
7175 046262 012760 000013 000000      MOV    #13, RMCS1(RO)  ; RELEASE THE DRIVE
7176 046270 113760 001226 000010      MOVB   PORTB, RMCS2(RO); SELECT PORT B
7177 046276 012760 177777 000014      MOV    #-1, RMER1(RO)  ; SET ERRORS TO FORCE ATTN BIT ON PORT B
7178 046304 005060 000014 000014      CLR    RMER1(RO)       ; CLEAR THE ERRORS
7179 046310 113760 001224 000010      MOVB   PORTA, RMCS2(RO); SELECT PORT A
7180 046316 005760 000012 000012      TST    RMDS1(RO)       ; WAIT FOR DRIVE TO RETURN TO NEUTRAL
7181 046322 001775 000012 000012      BEQ    IS              ; BR IF STILL SEIZED BY PORT B
7182 046324 012737 000016 001122      MOV    #RMAS, $BDAOR   ; FORM ADDRESS OF ATTN REG IF ERROR
7183 046332 060037 001122 001122      ADD    RO, $BDAOR      ; ADD THE ADDRESS BASE
7184 046336 013737 001236 001124      MOV    ASR1, $GDDAT    ; GOOD DATA FOR ERROR MESSAGE
7185 046344 013737 001236 001166      MOV    ASR1, $TMP1     ; MAKE DATA COMPARE MASK
7186 046352 005137 001166 001166      COM    $TMP1           ; COMPLEMENT IT
7187 046356 012737 046412 001110      MOV    #25, $LPERR     ; LOAD LOOP ON ERROR ADDRESS
7188 046364 113760 001224 000010      MOVB   PORTA, RMCS2(RO); SELECT PORT A
7189 046372 013737 001224 001240      MOV    PORTA, PTNBR    ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7190 046400 013737 001224 001242      MOV    PORTA, SEIZPT   ; 'SEIZED' PORT ADDRESS
7191 046406 005060 000012 000012      CLR    RMDS1(RO)       ; SEIZE THE DRIVE THROUGH PORT A
7192 046412 016037 000016 001126      MOV    RMAS(RO), $BDDAT; GET THE CONTENTS OF THE ATTENTION REG
7193 046420 013737 001126 001164      MOV    $BDDAT, $TMP0   ; PUT CONTENTS INTO WORKING LOCATION
7194 046426 043737 001166 001164      BIC    $TMP1, $TMP0    ; CLEAR OTHER BITS
7195 046434 023737 001124 001164      CMP    $GDDAT, $TMP0   ; SEE IF ATTN BIT FOR DRIVE SET
7196 046442 001401 000000 000000      BEQ    3$             ; BR IF SET
7197 046444 104053 000000 000000      ERROR  53             ; REPORT THE ERROR
7198 046446
7199
7200                                     ;RELEASE THE DRIVE FROM PORT A
7201
7202 046446 113760 001224 000010      MOVB   PORTA, RMCS2(RO); SELECT PORT A
7203 046454 013737 001224 001240      MOV    PORTA, PTNBR    ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7204 046462 012760 000013 000000      MOV    #13, RMCS1(RO)  ; ISSUE RELEASE THROUGH PORT A
7205
7206                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7207
7208 046470 005037 001254 000012 001122      CLR    RELERR          ; CLEAR THE 'RELEASE ERROR' INDICATOR
7209 046474 012737 000012 001122      MOV    #RMDS1, $BDAOR  ; FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7210 046502 060037 001122 001122      ADD    RO, $BDAOR      ; ADD THE I/O BASE ADDRESS
7211 046506 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ; COMPARISON CONSTANT
7212 046514 113760 001224 000010      MOVB   PORTA, RMCS2(RO); SELECT PORT A.
7213 046522 016037 000012 001170      MOV    RMDS1(RO), $TMP2; GET THE DRIVE STATUS REGISTER FROM PORT A.
7214 046530 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ; CLEAR DONT CARES
7215 046536 013737 001170 001164      MOV    $TMP2, $TMP0    ; COPY IT INTO 'TMP0'
7216 046544 042737 100100 001164      BIC    #ATA!VV, $TMP0  ; CLEAR PORT DEPENDENT BITS FROM THE COPY
7217 046552 113760 001226 000010      MOVB   PORTB, RMCS2(RO); SELECT PORT B.
7218 046560 016037 000012 001172      MOV    RMDS1(RO), $TMP3; GET THE DRIVE STATUS REGISTER FROM PORT B.
7219 046566 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ; CLEAR DONT CARES
7220 046574 013737 001172 001166      MOV    $TMP3, $TMP1    ; COPY IT INTO 'TMP1'
7221 046602 042737 100100 001166      BIC    #ATA!VV, $TMP1  ; CLEAR PORT DEPENDENT BITS FROM THE COP.

```

E11

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 134
T35 PORT 'B' ALTERNATE ATTENTION PATH TEST

SEQ 0134

```

7222 046610 023737 001164 001166      CMP      $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7223 046616 001006                BNE      64$             ;BR IF NOT
7224 046620 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7225 046624 001045                BNE      66$             ;BR IF NOT
7226 046626 104046                ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7227 046630 000137 047030                JMP      68$             ;BYPASS THE REST OF THE CHECKS
7228 046634 013737 001170 001126 64$:      MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7229 046642 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7230 046650 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B.
7231 046656 005737 001164                TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7232 046662 001414                BEQ      65$             ;BR IF ZERO
7233 046664 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7234 046672 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
7235 046700 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A.
7236 046706 005737 001166                TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7237 046712 001012                BNE      66$             ;BR IF NOT
7238 046714 012737 177777 001254 65$:      MOV      #-1,RELEA      ;SET 'RELEASE ERROR' INDICATOR
7239 046722 012760 000011 000000      MOV      #11,RMCS1(RO)  ;CLEAR THE DRIVE
7240 046730 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
7241 046736 104026                ERROR    26             ;TYPE ERROR MESSAGE 26
7242 046740 013737 001170 001126 66$:      MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7243 046746 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
7244 046754 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
7245 046762 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;ALL BITS OK ?
7246 046770 001401                BEQ      67$             ;BR IF OK FROM PORT A.
7247 046772 104007                ERROR    7              ;REPORT ERROR
7248 046774 013737 001172 001126 67$:      MOV      $TMP3,$BDDAT     ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7249 047002 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
7250 047010 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
7251 047016 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;SEE IF READ OK FROM PORT B.
7252 047024 001401                BEQ      68$             ;BR IF OK
7253 047026 104007                ERROR    7              ;REPORT ERROR
7254 047030 000240 68$:      NOP
7255 047032 000004                SCOPE

```

```

7256
7257 ;:*****
7258
7259 ;:*****
7260
7261 ;*TEST 36      SET ATTENTION 'A' BY COMMAND TEST
7262 ;*
7263 ;*
7264 ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
7265 ;*COMMAND.
7266 ;*
7267 ;* A.  ISSUE A OFFSET COMMAND THROUGH PORT 'A'.
7268 ;*
7269 ;* B.  WAIT FOR THE OFFSET COMMAND TO COMPLETE ('DRY' TO BECOME
7270 ;* '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
7271 ;* THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
7272 ;*
7273 ;* C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
7274 ;* TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7275 ;*
7276 ;:*****
7277 †ST36:

```

047034

F11

CZRMGB0 RM03/2 DL POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 135
T36 SET ATTENTION 'A' BY COMMAND TEST

SEQ 0135

```

7278 047034 005737 001300          TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7279 047040 001406          BEQ    25              ;BR IF NOT
7280 047042 100002          BPL    15              ;BR IF JUST ENTERED TEST
7281 047044 000137 002676          JMP    EXEC           ;RETURN & GET NEXT TEST NUMBER
7282 047050 012737 177777 001300 15:    MOV    #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
7283 047056 112737 000036 001102 25:    MOVB  #36,$STNM      ;TEST NUMBER
7284 047064 012737 047106 001106    MOV    #TEST36,$LPADR ;LOAD LOOP ON TEST ADDRESS
7285 047072 012737 047106 001110    MOV    #TEST36,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7286 047100 012737 000031 001176    MOV    #25,$TIMES    ;DO 25 ITERATIONS
7287 047106 012706 001100    TEST36: MOV #STACK,SP ;LOAD THE STACK POINTER
7288
7289          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7290
7291 047112 113760 001224 000010    MOVB  PORTA,RMCS2(RO) ;SELECT PORT #A
7292 047120 005060 000012          CLR    RMDS1(RO)      ;SEIZE THE DRIVE
7293 047124 012760 000011 000000    MOV    #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7294 047132 012760 000013 000000    MOV    #13,RMCS1(RO) ;RELEASE THE DRIVE
7295 047140 113760 001226 000010    MOVB  PORTB,RMCS2(RO) ;SELECT PORT #B
7296 047146 005060 000012          CLR    RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
7297 047152 012760 000011 000000    MOV    #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7298 047160 012760 000013 000000    MOV    #13,RMCS1(RO) ;RELEASE THE DRIVE
7299 047166 113760 001224 000010    MOVB  PORTA,RMCS2(RO) ;SELECT PORT A
7300 047174 013737 001224 001240    MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7301 047202 013737 001224 001242    MOV    PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
7302
7303          ;*****
7304          ;DO A OFFSET THROUGH PORT A
7305
7306 047210 012760 000015 000000    MOV    #15,RMCS1(RO) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT A
7307
7308          ;*****
7309          ;WAIT FOR DRIVE TO FINISH
7310
7311 047216 032760 000200 000012    BIT    #DRY,RMDS1(RO) ;WAIT FOR DRIVE TO FINISH
7312 047224 001774          BEQ    -6              ;BR IF NOT FINISHED
7313
7314          ;*****
7315          ;CONFIRM THAT ATTENTION IS SET FOR PORT A
7316
7317 047226 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
7318 047232 016037 000012 001126    MOV    RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
7319 047240 012737 000012 001122    MOV    #RMDS1,$BODADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7320 047246 060037 001122          ADD    RO,$BODADR     ;ADD RHI1 BASE ADDRESS
7321 047252 012737 100000 001124    MOV    #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
7322 047260 013737 001126 001164    MOV    $BODAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
7323 047266 042737 077777 001164    BIC    #ICATA,$TMP0   ;SAVE SPECIFIED BITS
7324 047274 023737 001124 001164    CMP    $GDDAT,$TMP0   ;COMPARE THE BITS
7325 047302 001414          BEQ    64$            ;BR IF OK
7326 047304 013737 001126 001174    MOV    $BODAT,$TMP4   ;COPY 'BAD DATA'
7327 047312 042737 100000 001174    BIC    #ATA,$TMP4     ;CLEAR THE MASKED BITS
7328 047320 053737 001174 001124    BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
7329 047326 104032          ERROR 32              ;TYPE MESSAGE 32
7330 047330 005137 001250    COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
7331 047334 000240    64$:  NOP
7332
7333          ;RELEASE THE DRIVE FROM PORT A

```


G11

CZRMGB0 RMO3 2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 136
T36 SET ATTENTION 'A' BY COMMAND TEST

SEQ 0136

```

7334
7335 047336 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
7336 047344 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7337 047352 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7338
7339                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7340
7341 047360 005037 001254                                     CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7342 047364 012737 000012 001122      MOV    #RMDS1, $BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7343 047372 060037 001122                                     ADD    RO, $BDADR ;ADD THE I/O BASE ADDRESS
7344 047376 012737 011700 001124      MOV    #MCL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
7345 047404 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
7346 047412 016037 000012 001170      MOV    RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7347 047420 042737 024001 001170      BIC    #PIP:WRL:OM,$TMP2 ;CLEAR DONT CARES
7348 047426 013737 001170 001164      MOV    $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
7349 047434 042737 100100 001164      BIC    #ATA:VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7350 047442 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
7351 047450 016037 000012 001172      MOV    RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7352 047456 042737 024001 001172      BIC    #PIP:WRL:OM,$TMP3 ;CLEAR DONT CARES
7353 047464 013737 001172 001166      MOV    $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
7354 047472 042737 100100 001166      BIC    #ATA:VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7355 047500 023737 001164 001166      CMP    $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7356 047506 001006                                     BNE    66$ ;BR IF NOT
7357 047510 005737 001164                                     TST    $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7358 047514 001045                                     BNE    68$ ;BR IF NOT
7359 047516 104046      ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7360 047520 000137 047704      JMP    70$ ;BYPASS THE REST OF THE CHECKS
7361 047524 013737 001170 001126 66$:      MOV    $TMP2, $BDADR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7362 047532 013737 001226 001240      MOV    PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7363 047540 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
7364 047546 005737 001164      TST    $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7365 047552 001414      BEQ    67$ ;BR IF ZERO
7366 047554 013737 001224 001240      MOV    PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7367 047562 013737 001172 001126      MOV    $TMP3, $BDADR ;'BAD DATA' FOR ERROR TYPE OUT
7368 047570 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
7369 047576 005737 001166      TST    $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7370 047602 001012      BNE    68$ ;BR IF NOT
7371 047604 012737 177777 001254 67$:      MOV    #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
7372 047612 012760 000011 000000      MOV    #11, RMCS1(RO) ;CLEAR THE DRIVE
7373 047620 012760 000013 000000      MOV    #13, RMCS1(RO) ;RELEASE THE DRIVE
7374 047626 104026      ERROR    26 ;TYPE ERROR MESSAGE 26
7375 047630 013737 001170 001126 68$:      MOV    $TMP2, $BDADR ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7376 047636 013737 001224 001240      MOV    PORTA, PTNBR ;CHANGE PORT NUMBER
7377 047644 023737 001124 001126      CMP    $GDDAT, $BDADR ;ALL BITS OK ?
7378 047652 001401      BEQ    69$ ;BR IF OK FROM PORT A.
7379 047654 104007      ERROR    7 ;REPORT ERROR
7380 047656 013737 001172 001126 69$:      MOV    $TMP3, $BDADR ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7381 047664 013737 001226 001240      MOV    PORTB, PTNBR ;CHANGE PORT NUMBER
7382 047672 023737 001124 001126      CMP    $GDDAT, $BDADR ;SEE IF READ OK FROM PORT B.
7383 047700 001401      BEQ    70$ ;BR IF OK
7384 047702 104007      ERROR    7 ;REPORT ERROR
7385 047704 000240      NOP
7386 047706 113760 001226 000010 70$:      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
7387 047714 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7388
7389
;*****

```

H11

C2RMGB0 RM03/2 DU POR LGC 1
C2RMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 137
T36 SET ATTENTION 'A' BY COMMAND TEST

SEQ 0137

:CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

7390										
7391										
7392	047722	005037	001250			CLR	CKERR		;CLEAR THE 'CHECK ERROR' INDICATOR	
7393	047726	016037	000012	001126		MOV	RMDS1(R0), \$BDDAT		;GET CONTENTS OF RMDS1	
7394	047734	012737	000012	001122		MOV	#RMDS1, \$BDADR		;FORM REGISTER ADDRESS OF ERROR MESSAGE	
7395	047742	060037	001122			ADD	R0, \$BDADR		;ADD R011 BASE ADDRESS	
7396	047746	005037	001124			CLR	\$GDDAT		;WHAT REGISTER SHOULD BE	
7397	047752	013737	001126	001164		MOV	\$BDDAT, \$TMP0		;MOVE REGISTER CONTENTS TO '\$TMP0'	
7398	047760	042737	077777	001164		BIC	#ICATA, \$TMP0		;SAVE SPECIFIED BITS	
7399	047766	023737	001124	001164		CMP	\$GDDAT, \$TMP0		;COMPARE THE BITS	
7400	047774	001414				BEQ	71\$;BR IF OK	
7401	047776	013737	001126	001174		MOV	\$BDDAT, \$TMP4		;COPY 'BAD DATA'	
7402	050004	042737	100000	001174		BIC	#ATA, \$TMP4		;CLEAR THE MASKED BITS	
7403	050012	053737	001174	001124		BIS	\$TMP4, \$GDDAT		; 'OR' WITH GOOD DATA FOR TYPEOUT	
7404	050020	104032				ERROR	32		;TYPE MESSAGE 32	
7405	050022	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR	
7406	050026	000240				71\$:	NOB			

::*****
SCOPE ;LOOP ?

::*****
*TEST 37 SET ATTENTION 'B' BY COMMAND TEST

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

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

::*****
↑ST37:

7427	050032									
7428	050032	005737	001300			TST	KYBCTL		;PERFORMING ONLY SINGLE TESTS ?	
7429	050036	001406				BEQ	2\$;BR IF NOT	
7430	050040	100002				BPL	1\$;BR IF JUST ENTERED TEST	
7431	050042	000137	002676			JMP	EXEC		;RETURN & GET NEXT TEST NUMBER	
7432	050046	012737	177777	001300	1\$:	MOV	#-1, KYBCTL		;SET SINGLE TEST INDICATOR	
7433	050054	112737	000037	001102	2\$:	MOVB	#37, \$STNM		;TEST NUMBER	
7434	050062	012737	050104	001106		MOV	#TEST37, \$LPADR		;LOAD LOOP ON TEST ADDRESS	
7435	050070	012737	050104	001110		MOV	#TEST37, \$LPERR		;LOAD LOOP ON ERROR ADDRESS	
7436	050076	012737	000031	001176		MOV	#25, \$TIMES		;DO 25 ITERATIONS	
7437	050104	012706	001100		TEST37:	MOV	#STACK, SP		;LOAD THE STACK POINTER	

;CLEAR ATTENTION BITS FOR BOTH PORTS

7440										
7441	050110	113760	001224	000010		MOVB	PORTA, RMCS2(R0)		;SELECT PORT #A	
7442	050116	005060	000012			CLR	RMDS1(R0)		;SEIZE THE DRIVE	
7443	050122	012760	000011	000000		MOV	#11, RMCS1(R0)		;ISSUE DRIVE CLEAR	
7444	050130	012760	000013	000000		MOV	#13, RMCS1(R0)		;RELEASE THE DRIVE	
7445	050136	113760	001226	000010		MOVB	PORTB, RMCS2(R0)		;SELECT PORT #B	

```

7446 050144 005060 000012          CLR      RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
7447 050150 012760 000011 000000    MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
7448 050156 012760 000013 000000    MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
7449 050164 113760 001226 000010    MOV      PORTB,RMCS2(RO);SELECT PORT B
7450 050172 013737 001226 001240    MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7451 050200 013737 001226 001242    MOV      PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
7452
7453 ;:*****
7454 ;DO A OFFSET THROUGH PORT B
7455
7456 050206 012760 000015 000000    MOV      #15,RMCS1(RO) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT B
7457
7458 ;:*****
7459 ;WAIT FOR DRIVE TO FINISH
7460
7461 050214 032760 000200 000012    BIT      #DRY,RMDS1(RO);WAIT FOR DRIVE TO FINISH
7462 050222 001774          BEQ      -6            ;BR IF NOT FINISHED
7463
7464 ;:*****
7465 ;CONFIRM THAT ATTENTION IS SET FOR PORT B
7466
7467 050224 005037 001250          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
7468 050230 016037 000012 001126    MOV      RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
7469 050236 012737 000012 001122    MOV      #RMDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7470 050244 060037 001122          ADD      RO,$BDAOR    ;ADD RHI1 BASE ADDRESS
7471 050250 012737 100000 001124    MOV      #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
7472 050256 013737 001126 001164    MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
7473 050264 042737 077777 001164    BIC      #ICATA,$STMP0 ;SAVE SPECIFIED BITS
7474 050272 023737 001124 001164    CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
7475 050300 001414          BEQ      645         ;BR IF OK
7476 050302 013737 001126 001174    MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
7477 050310 042737 100000 001174    BIC      #ATA,$STMP4  ;CLEAR THE MASKED BITS
7478 050316 053737 001174 001124    BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7479 050324 104032          ERROR   32          ;TYPE MESSAGE 32
7480 050326 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
7481 050332 000240          645:  NOP
7482
7483 ;RELEASE THE DRIVE FROM PORT B
7484
7485 050334 113760 001226 000010    MOV      PORTB,RMCS2(RO);SELECT PORT B
7486 050342 013737 001226 001240    MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7487 050350 012760 000013 000000    MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
7488
7489 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7490
7491 050356 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
7492 050362 012737 000012 001122    MOV      #RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7493 050370 060037 001122          ADD      RO,$BDAOR  ;ADD THE I/O BASE ADDRESS
7494 050374 012737 011700 001124    MOV      #MOL!PGM!OPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7495 050402 113760 001224 000010    MOV      PORTA,RMCS2(RO);SELECT PORT A.
7496 050410 016037 000012 001170    MOV      RMDS1(RO), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7497 050416 042737 024001 001170    BIC      #PIP!WAL!OM,$STMP2 ;CLEAR DONT CARES
7498 050424 013737 001170 001164    MOV      $STMP2,$STMP0 ;COPY IT INTO '$STMP0'
7499 050432 042737 100100 001164    BIC      #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7500 050440 113760 001226 000010    MOV      PORTB,RMCS2(RO);SELECT PORT B.
7501 050446 016037 000012 001172    MOV      RMDS1(RO), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

```

7502 050454 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7503 050462 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7504 050470 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7505 050476 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7506 050504 001006 BNE 66$ ;BR IF NOT
7507 050506 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7508 050512 001045 BNE 68$ ;BR IF NOT
7509 050514 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7510 050516 000137 050702 JMP 70$ ;BYPASS THE REST OF THE CHECKS
7511 050522 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7512 050530 013737 001226 001240 MOV PORTB,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7513 050536 113760 001226 000010 MOVB PORTB,$RMCS2(RO) ;SELECT PORT B.
7514 050544 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7515 050550 001414 BEQ 67$ ;BR IF ZERO
7516 050552 013737 001224 001240 MOV PORTA,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7517 050560 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7518 050566 113760 001224 000010 MOVB PORTA,$RMCS2(RO) ;SELECT PORT A.
7519 050574 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7520 050600 001012 BNE 68$ ;BR IF NOT
7521 050602 012737 177777 001254 67$: MOV #-1,$REERR ;SET 'RELEASE ERROR' INDICATOR
7522 050610 012760 000011 000000 MOV #11,$RMCS1(RO) ;CLEAR THE DRIVE
7523 050616 012760 000013 000000 MOV #13,$RMCS1(RO) ;RELEASE THE DRIVE
7524 050624 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
7525 050626 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7526 050634 013737 001224 001240 MOV PORTA,$PTNBR ;CHANGE PORT NUMBER
7527 050642 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
7528 050650 001401 BEQ 69$ ;BR IF OK FROM PORT A.
7529 050652 104007 ERROR 7 ;REPORT ERROR
7530 050654 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7531 050662 013737 001226 001240 MOV PORTB,$PTNBR ;CHANGE PORT NUMBER
7532 050670 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
7533 050676 001401 BEQ 70$ ;BR IF OK
7534 050700 104007 ERROR 7 ;REPORT ERROR
7535 050702 000240 70$: NOP
7536 050704 113760 001224 000010 MOVB PORTA,$RMCS2(RO) ;SELECT PORT A
7537 050712 013737 001224 001240 MOV PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7538
7539 ;*****
7540 ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A
7541
7542 050720 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7543 050724 016037 000012 001126 MOV RMDS1(RO),$BDDAT ;GET CONTENTS OF RMDS1
7544 050732 012737 000012 001122 MOV #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7545 050740 060037 001122 ADD RO,$BDDADR ;ADD RHI1 BASE ADDRESS
7546 050744 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7547 050750 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7548 050756 042737 077777 001164 BIC #!CATA,$TMP0 ;SAVE SPECIFIED BITS
7549 050764 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
7550 050772 001414 BEQ 71$ ;BR IF OK
7551 050774 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
7552 051002 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
7553 051010 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7554 051016 104032 ERROR 32 ;TYPE MESSAGE 32
7555 051020 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7556 051024 000240 71$: NOP
7557

```

```

7558 ;*****
7559 051026 000004 ; SCOPE ; LOOP ?
7560
7561 ;*****
7562 ;
7563 ;
7564 ;*VERIFY THAT A CHANGE IN UNIT READY SETS THE ATTENTION
7565 ;* FOR BOTH PORTS.
7566 ;
7567 ;*THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.
7568 ;
7569 ;*****
7570 ;
7571 ;*****
7572 ;
7573 ;*VERIFY THAT ATTENTION SETS WHEN THE DRIVE SWITCHES AFTER
7574 ;*BEING RELEASED.
7575 ;
7576 ;*THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"
7577 ;
7578 ;*****
7579 ;
7580 ;*****
7581 ;
7582 ;*TEST 40 PORT 'A' SET VOLUME VALID TEST
7583 ;*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
7584 ;
7585 ;* A. WITH PORT 'A' SELECTED, RESET AND SET "UNIT READY"
7586 ;* STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
7587 ;* IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
7588 ;* ATTENTION IS SET.
7589 ;
7590 ;* B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
7591 ;* COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
7592 ;* VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
7593 ;* IS SET.
7594 ;
7595 ;* C. RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR
7596 ;* PORT 'B'. VERIFY THAT ATTENTION IS STILL SET AND THAT
7597 ;* VOLUME VALID IS STILL RESET.
7598 ;
7599 ;* D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
7600 ;* PORT 'B' THEN RELEASE PORT 'B'.
7601 ;*****
7602 051030 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ^
7603 051030 005737 001300 BEQ 25 ;BR IF NOT
7604 051034 001406 BPL 15 ;BR IF JUST ENTERED TEST
7605 051036 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
7606 051040 000137 002676 MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
7607 051044 012737 177777 001300 15: ;TEST NUMBER
7608 051052 112737 000040 001102 25: MOV8 #40,$STNM ;LOAD LOOP ON TEST ADDRESS
7609 051060 012737 051102 001106 MOV #TEST40,$LPADR ;LOAD LOOP ON ERROR ADDRESS
7610 051066 012737 051102 001110 MOV #TEST40,$LPERR ;DO 25. ITERATIONS
7611 051074 012737 000031 001176 MOV #25,$TIMES ;LOAD THE STACK POINTER
7612 051102 012706 001100 TEST40: MOV #STACK,SP
7613

```

```

7614
7615
7616
7617
7618
7619
7620
7621 051106 113760 001224 000010      MOV      PORTA, RMCS2(RO) ; SELECT PORT A
7622 051114 013737 001224 001242      MOV      PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
7623 051122 012760 000001 000024      MOV      #DMD, RMMR1(RO) ; WRITE #DMD INTO RMMR1
7624 051130 013737 001226 001244      MOV      PORTA, OPPRT ; 'OPPOSITE' PORT ADDRESS
7625 051136 012760 001001 000024      MOV      #DMD, RMMR1(RO) ; SET UNIT READY
7626 051144 012760 000000 000024      MOV      #0, RMMR1(RO) ; RESET DIAGNOSTIC MODE
7627
7628
7629
7630
7631 051152 005037 001250
7632 051156 016037 000012 001126
7633 051164 012737 000012 001122
7634 051172 060037 001122
7635 051176 012737 100000 001124
7636 051204 013737 001126 001164
7637 051212 042737 077777 001164
7638 051220 023737 001124 001164
7639 051226 001414
7640 051230 013737 001126 001174
7641 051236 042737 100000 001174
7642 051244 053737 001174 001124
7643 051252 104064
7644 051254 005137 001250
7645 051260 000240
7646 051262 005037 001250
7647 051266 016037 000012 001126
7648 051274 012737 000012 001122
7649 051302 060037 001122
7650 051306 005037 001124
7651 051312 013737 001126 001164
7652 051320 042737 177677 001164
7653 051326 023737 001124 001164
7654 051334 001414
7655 051336 013737 001126 001174
7656 051344 042737 000100 001174
7657 051352 053737 001174 001124
7658 051360 104065
7659 051362 005137 001250
7660 051366 000240
7661
7662 051370 012760 000011 000000
7663 051376 012760 000021 000000
7664
7665
7666 051404 005037 001250
7667 051410 016037 000012 001126
7668 051416 012737 000012 001122
7669 051424 060037 001122

```

```

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

; SEIZE THE DRIVE THROUGH PORT A

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

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

; ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A
MOV      #11, RMCS1(RO) ; DRIVE CLEAR
MOV      #21, RMCS1(RO) ; READ IN PRESET

; VERIFY ATA=0 AND VV=1 FOR PORT A
CLR      CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV      RMDS1(RO), $BDDAT ; GET CONTENTS OF RMDS1
MOV      #RMDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD      RO, $BDAOR ; ADD RH11 BASE ADDRESS

```

M11

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 142
T40 PORT 'A' 5-T VOLUME VALID TEST

SEG 0142

7670	051430	012737	000100	001124	MOV	#VV,\$GDDAT	;WHAT REGISTER SHOULD BE
7671	051436	013737	001126	001164	MOV	\$BDDAT,\$STMP0	;MOVE REGISTER CONTENTS TO '\$STMP0'
7672	051444	042737	177677	001164	BIC	#1CVV,\$STMP0	;SAVE SPECIFIED BITS
7673	051452	023737	001124	001164	CMP	\$GDDAT,\$STMP0	;COMPARE THE BITS
7674	051460	001414			BEQ	70\$;BR IF OK
7675	051462	013737	001126	001174	MOV	\$BDDAT,\$STMP4	;COPY 'BAD DATA'
7676	051470	042737	000100	001174	BIC	#VV,\$STMP4	;CLEAR THE MASKED BITS
7677	051476	053737	001174	001124	BIS	\$STMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7678	051504	104013			ERROR	13	;TYPE MESSAGE 13
7679	051506	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
7680	051512	000240			NOP		
7681	051514	005037	001250		70\$: CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
7682	051520	016037	000012	001126	MOV	\$RMS1(\$RO),\$BDDAT	;GET CONTENTS OF RMS1
7683	051526	012737	000012	001122	MOV	\$RMS1,\$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
7684	051534	060037	001122		ADD	\$RO,\$BDAOR	;ADD R11 BASE ADDRESS
7685	051540	012737	011700	001124	MOV	#MOL:PGM:DPR:DRY:VV,\$GDDAT	;WHAT REGISTER SHOULD BE
7686	051546	013737	001126	001164	MOV	\$BDDAT,\$STMP0	;MOVE REGISTER CONTENTS TO '\$STMP0'
7687	051554	042737	024007	001164	BIC	#1C153770,\$STMP0	;SAVE SPECIFIED BITS
7688	051562	023737	001124	001164	CMP	\$GDDAT,\$STMP0	;COMPARE THE BITS
7689	051570	001414			BEQ	72\$;BR IF OK
7690	051572	013737	001126	001174	MOV	\$BDDAT,\$STMP4	;COPY 'BAD DATA'
7691	051600	042737	153770	001174	BIC	#153770,\$STMP4	;CLEAR THE MASKED BITS
7692	051606	053737	001174	001124	BIS	\$STMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
7693	051614	104010			ERROR	10	;REPORT THE ERROR
7694	051616	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
7695	051622	000240			72\$: NOP		
7696							
7697							
7698							
7699							
7700							
7701							
7702	051624	113760	001224	000010	MOV	PORTA,\$RMS2(\$RO)	;SELECT PORT A
7703	051632	013737	001224	001240	MOV	PORTA,\$PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7704	051640	012760	000013	000000	MOV	#13,\$RMS1(\$RO)	;ISSUE RELEASE THROUGH PORT A
7705							
7706							
7707							
7708	051646	005037	001254		CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
7709	051652	012737	000012	001122	MOV	\$RMS1,\$BDAOR	;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
7710	051660	060037	001122		ADD	\$RO,\$BDAOR	;ADD THE I/O BASE ADDRESS
7711	051664	012737	011600	001124	MOV	#MOL:PGM:DPR:DRY:\$GDDAT	;COMPARISON CONSTANT
7712	051672	113760	001224	000010	MOV	PORTA,\$RMS2(\$RO)	;SELECT PORT A.
7713	051700	016037	000012	001170	MOV	\$RMS1(\$RO),\$STMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
7714	051706	042737	024001	001170	BIC	#PIP:WRL:OM,\$STMP2	;CLEAR DONT CARES
7715	051714	013737	001170	001164	MOV	\$STMP2,\$STMP0	;COPY IT INTO '\$STMP0'
7716	051722	042737	100100	001164	BIC	#ATA:VV,\$STMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7717	051730	113760	001226	000010	MOV	PORTB,\$RMS2(\$RO)	;SELECT PORT B.
7718	051736	016037	000012	001172	MOV	\$RMS1(\$RO),\$STMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
7719	051744	042737	024001	001172	BIC	#PIP:WRL:OM,\$STMP3	;CLEAR DONT CARES
7720	051752	013737	001172	001166	MOV	\$STMP3,\$STMP1	;COPY IT INTO '\$STMP1'
7721	051760	042737	100100	001166	BIC	#ATA:VV,\$STMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
7722	051766	023737	001164	001166	CMP	\$STMP0,\$STMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7723	051774	001006			BNE	74\$;BR IF NOT
7724	051776	005737	001164		TST	\$STMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7725	052002	001037			BNE	76\$;BR IF NOT

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

;RELEASE THE DRIVE FROM PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

N11

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 143
T40 PORT 'A' SET VOLUME VALID TEST

SEQ 0143

7726	052004	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7727	052006	000137	052172			JMP	78\$;BYPASS THE REST OF THE CHECKS
7728	052012	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7729	052020	013737	001226	001240		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7730	052026	113760	001226	000010		MOV	PORTB, RMCS2(RO)		;SELECT PORT B.
7731	052034	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
7732	052040	001414				BEQ	75\$;BR IF ZERO
7733	052042	013737	001224	001240		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7734	052050	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
7735	052056	113760	001224	000010		MOV	PORTA, RMCS2(RO)		;SELECT PORT A.
7736	052064	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
7737	052070	001004				BNE	76\$;BR IF NOT
7738	052072	012737	177777	001254	75\$:	MOV	#-1,RELERR		;SET 'RELEASE ERROR' INDICATOR
7739	052100	104022				ERROR	22		;TYPE ERROR MESSAGE 22
7740	052102	013737	001170	001126	76\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7741	052110	013737	001224	001240		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER
7742	052116	042737	100100	001126		SIC	#ATA!VV,\$BDDAT		;DON'T CHECK ATTN BIT OR VV BIT
7743	052124	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;ALL BITS OK ?
7744	052132	001401				BEQ	77\$;BR IF OK FROM PORT A.
7745	052134	104007				ERROR	7		;REPORT ERROR
7746	052136	013737	001172	001126	77\$:	MOV	\$TMP3,\$BDDAT		;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
7747	052144	013737	001226	001240		MOV	PORTB,PTNBR		;CHANGE PORT NUMBER
7748	052152	042737	100100	001126		BIC	#ATA!VV,\$BDDAT		;DON'T CHECK ATTN BIT OR VV BIT
7749	052160	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;SEE IF READ OK FROM PORT B.
7750	052166	001401				BEQ	78\$;BR IF OK
7751	052170	104007				ERROR	7		;REPORT ERROR
7752	052172	000240			78\$:	NOP			
7753	052174	113760	001226	000010		MOV	PORTB, RMCS2(RO)		;SELECT PORT B
7754	052202	013737	001226	001240		MOV	PORTB,PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7755	052210	005037	001250			CLR	CKERR		;CLEAR THE 'CHECK ERROR' INDICATOR
7756	052214	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT		;GET CONTENTS OF RMDS1
7757	052222	012737	000012	001122		MOV	#RMDS1,\$BDAOR		;FORM REGISTER ADDRESS OF ERROR MESSAGE
7758	052230	060037	001122			ADD	RO,\$BDAOR		;ADD RH11 BASE ADDRESS
7759	052234	012737	100000	001124		MOV	#ATA,\$GDDAT		;WHAT REGISTER SHOULD BE
7760	052242	013737	001126	001164		MOV	\$BDDAT,\$TMP0		;MOVE REGISTER CONTENTS TO '\$TMP0'
7761	052250	042737	077777	001164		BIC	#ICATA,\$TMP0		;SAVE SPECIFIED BITS
7762	052256	023737	001124	001164		CMP	\$GDDAT,\$TMP0		;COMPARE THE BITS
7763	052264	001414				BEQ	79\$;BR IF OK
7764	052266	013737	001126	001174		MOV	\$BDDAT,\$TMP4		;COPY 'BAD DATA'
7765	052274	042737	100000	001174		BIC	#ATA,\$TMP4		;CLEAR THE MASKED BITS
7766	052302	053737	001174	001124		BIS	\$TMP4,\$GDDAT		; 'OR' WITH GOOD DATA FOR TYPEOUT
7767	052310	104064				ERROR	64		;TYPE MESSAGE 64
7768	052312	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
7769	052316	000240			79\$:	NOP			
7770	052320	005037	001250			CLR	CKERR		;CLEAR THE 'CHECK ERROR' INDICATOR
7771	052324	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT		;GET CONTENTS OF RMDS1
7772	052332	012737	000012	001122		MOV	#RMDS1,\$BDAOR		;FORM REGISTER ADDRESS OF ERROR MESSAGE
7773	052340	060037	001122			ADD	RO,\$BDAOR		;ADD RH11 BASE ADDRESS
7774	052344	005037	001124			CLR	\$GDDAT		;WHAT REGISTER SHOULD BE
7775	052350	013737	001126	001164		MOV	\$BDDAT,\$TMP0		;MOVE REGISTER CONTENTS TO '\$TMP0'
7776	052356	042737	177677	001164		BIC	#ICVV,\$TMP0		;SAVE SPECIFIED BIT
7777	052364	023737	001124	001164		CMP	\$GDDAT,\$TMP0		;COMPARE THE BITS
7778	052372	001414				BEQ	BIS		;BR IF OK
7779	052374	013737	001126	001174		MOV	\$BDDAT,\$TMP4		;COPY 'BAD DATA'
7780	052402	042737	000100	001174		BIC	#VV,\$TMP4		;CLEAR THE MASKED BITS
7781	052410	053737	001174	001124		BIS	\$TMP4,\$GDDAT		; 'OR' WITH GOOD DATA FOR TYPEOUT

7782	052416	104065				ERROR	65		;TYPE MESSAGE 65
7783	052420	005137	001250			COM	CKERR		;SET THE REGISTER COMPARE ERROR INDICATOR
7784	052424	000240				81\$:	NOP		
7785									
7786									;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B,
7787									;THEN RELEASE PORT B
7788	052426	012760	000011	000000		MOV	#11,RMCS1(RO)		;DRIVE CLEAR
7789	052434	012760	000021	000000		MOV	#21,RMCS1(RO)		;READ IN PRESET
7790									
7791									;RELEASE THE DRIVE FROM PORT B
7792									
7793	052442	113760	001226	000010		MOV	PORTB,RMCS2(RO)		;SELECT PORT B
7794	052450	013737	001226	001240		MOV	PORTB,PTNBR		;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7795	052456	012760	000013	000000		MOV	#13,RMCS1(RO)		;ISSUE RELEASE THROUGH PORT B
7796									
7797									;VERIFY THAT THE DRIVE IS IN NEUTRAL
7798									
7799	052464	005037	001254			CLR	RELEERR		;CLEAR THE 'RELEASE ERROR' INDICATOR
7800	052470	012737	000012	001122		MOV	#RMDS1,\$BDDADR		;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
7801	052476	060037	001122			ADD	RO,\$BDDADR		;ADD THE I/O BASE ADDRESS
7802	052502	012737	011600	001124		MOV	#MOL:PGM:DPR:DRY,\$GDDAT		;COMPARISON CONSTANT
7803	052510	113760	001224	000010		MOV	PORTA,RMCS2(RO)		;SELECT PORT A.
7804	052516	016037	000012	001170		MOV	RMDS1(RO),\$TMP2		;GET THE DRIVE STATUS REGISTER FROM PORT A.
7805	052524	042737	024001	001170		BIC	#PIP:WRL:OM,\$TMP2		;CLEAR DONT CARES
7806	052532	013737	001170	001164		MOV	\$TMP2,\$TMP0		;COPY IT INTO '\$TMP0'
7807	052540	042737	100100	001164		BIC	#ATA:VV,\$TMP0		;CLEAR PORT DEPENDENT BITS FROM THE COPY
7808	052546	113760	001226	000010		MOV	PORTB,RMCS2(RO)		;SELECT PORT B.
7809	052554	016037	000012	001172		MOV	RMDS1(RO),\$TMP3		;GET THE DRIVE STATUS REGISTER FROM PORT B.
7810	052562	042737	024001	001172		BIC	#PIP:WRL:OM,\$TMP3		;CLEAR DONT CARES
7811	052570	013737	001172	001166		MOV	\$TMP3,\$TMP1		;COPY IT INTO '\$TMP1'
7812	052576	042737	100100	001166		BIC	#ATA:VV,\$TMP1		;CLEAR PORT DEPENDENT BITS FROM THE COPY
7813	052604	023737	001164	001166		CMP	\$TMP0,\$TMP1		;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7814	052612	001006				BNE	83\$;BR IF NOT
7815	052614	005737	001164			TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7816	052620	001037				BNE	85\$;BR IF NOT
7817	052622	104046				ERROR	46		;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7818	052624	000137	053010			JMP	87\$;BYPASS THE REST OF THE CHECKS
7819	052630	013737	001170	001126	83\$:	MOV	\$TMP2,\$BDDAT		;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7820	052636	013737	001226	001240		MOV	PORTB,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7821	052644	113760	001226	000010		MOV	PORTB,RMCS2(RO)		;SELECT PORT B.
7822	052652	005737	001164			TST	\$TMP0		;SEE IF STATUS EQ 0 FROM PORT A.
7823	052656	001414				BEQ	84\$;BR IF ZERO
7824	052660	013737	001224	001240		MOV	PORTA,PTNBR		;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7825	052666	013737	001172	001126		MOV	\$TMP3,\$BDDAT		; 'BAD DATA' FOR ERROR TYPE OUT
7826	052674	113760	001224	000010		MOV	PORTA,RMCS2(RO)		;SELECT PORT A.
7827	052702	005737	001166			TST	\$TMP1		;SEE IF STATUS EQ ZERO FROM PORT B.
7828	052706	001004				BNE	85\$;BR IF NOT
7829	052710	012737	177777	001254	84\$:	MOV	#-1,RELEERR		;SET 'RELEASE ERROR' INDICATOR
7830	052716	104022				ERROR	22		;TYPE ERROR MESSAGE 22
7831	052720	013737	001170	001126	85\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RMDS1 READ
7832	052726	013737	001224	001240		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER
7833	052734	042737	100100	001126		BIC	#ATA:VV,\$BDDAT		;DON'T CHECK ATTN BIT OR VV BIT
7834	052742	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		;ALL BITS OK ?
7835	052750	001401				BEQ	86\$;BR IF OK FROM PORT A.
7836	052752	104007				ERROR	7		;REPORT ERROR
7837	052754	013737	001172	001126	86\$:	MOV	\$TMP3,\$BDDAT		;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.

```

7838 052762 013737 001226 001240
7839 052770 042737 100100 001126
7840 052776 023737 001124 001126
7841 053004 001401
7842 053006 104007
7843 053010 000240
7844
7845 053012 000004
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893

```

```

MOV PORTB,PTNBR ;CHANGE PORT NUMBER
BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
CMP $GDDAT,$ADDAT ;SEE IF READ OK FROM PORT B.
BEQ B7$ ;BR IF OK
ERROR 7 ;REPORT ERROR
B7$: NOP
50$: SCOPE

```

```

*****
*TEST 41 PORT 'B' SET VOLUME VALID TEST
*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
*
* A. WITH PORT 'B' SELECTED, RESET AND SET "UNIT READY"
* STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
* IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
* ATTENTION IS SET.
*
* B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
* COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
* VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
* IS SET.
*
* C. RELEASE THE DRIVE FROM PORT 'B' AND SELECT THE DRIVE FOR
* PORT 'A'. VERIFY THAT ATTENTION IS STILL SET AND THAT
* VOLUME VALID IS STILL RESET.
*
* D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
* PORT 'A' THEN RELEASE PORT 'A'.
*****

```

```

*****
*ST41:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ Z$ ;BR IF NOT
BPL I$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
I$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
Z$: MOVB #4,$STSTNM ;TEST NUMBER
MOV #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST41: MOV #STACK,SP ;LOAD THE STACK POINTER

```

```

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

```

```

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

```

```

7894
7895
7896
7897 053136 005037 001250
7898 053142 016037 000012 001126
7899 053150 012737 000012 001122
7900 053156 060037 001122
7901 053162 012737 100000 001124
7902 053170 013737 001126 001164
7903 053176 042737 077777 001164
7904 053204 023737 001124 001164
7905 053212 001414
7906 053214 013737 001126 001174
7907 053222 042737 100000 001174
7908 053230 053737 001174 001124
7909 053236 104064
7910 053240 005137 001250
7911 053244 000240 66$: NOP
7912 053246 005037 001250
7913 053252 016037 000012 001126
7914 053260 012737 000012 001122
7915 053266 060037 001122
7916 053272 005037 001124
7917 053276 013737 001126 001164
7918 053304 042737 177677 001164
7919 053312 023737 001124 001164
7920 053320 001414
7921 053322 013737 001126 001174
7922 053330 042737 000100 001174
7923 053336 053737 001174 001124
7924 053344 104065
7925 053346 005137 001250
7926 053352 000240 68$: NOP
7927
7928 ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B
7929 053354 012760 000011 000000
7930 053362 012760 000021 000000
7931
7932 ;VERIFY ATA=0 AND VV=1 FOR PORT B
7933 053370 075037 001250
7934 053374 016037 000012 001126
7935 053402 012737 000012 001122
7936 053410 060037 001122
7937 053414 012737 000100 001124
7938 053422 013737 001126 001164
7939 053430 042737 177677 001164
7940 053436 023737 001124 001164
7941 053444 001414
7942 053446 013737 001126 001174
7943 053454 042737 000100 001174
7944 053462 053737 001174 001124
7945 053470 104013
7946 053472 005137 001250
7947 053476 000240 70$: NOP
7948 053500 005037 001250
7949 053504 016037 000012 001126

```

E12

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 147
T41 PORT 'B' SET VOLUME VALID TEST

SEQ 0147

```

7950 053512 012737 000012 001122      MOV      #RMS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7951 053520 060037 001122      ADD      R0,$B0ADR ;ADD RHI1 BASE ADDRESS
7952 053524 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
7953 053532 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7954 053540 042737 024007 001164      BIC      #1C153770,$TMP0 ;SAVE SPECIFIED BITS
7955 053546 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
7956 053554 001414      BEQ      72$ ;BR IF OK
7957 053556 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
7958 053564 042737 153770 001174      BIC      #153770,$TMP4 ;CLEAR THE MASKED BITS
7959 053572 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7960 053600 104010      ERROR   10 ;REPORT THE ERROR
7961 053602 005137 001250      COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7962 053606 000240      72$:   NOP
7963
7964 ;RELEASE PORT B AND SELECT PORT A
7965 ;VERIFY THAT ATA=1 AND VV=0 FOR PORT A
7966
7967 ;RELEASE THE DRIVE FROM PORT B
7968
7969 053610 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
7970 053616 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7971 053624 012760 000013 000000      MOV      #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
7972
7973 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7974
7975 053632 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7976 053636 012737 000012 001122      MOV      #RMS1,$B0ADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
7977 053644 060037 001122      ADD      R0,$B0ADR ;ADD THE I/O BASE ADDRESS
7978 053650 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY!$GDDAT ;COMPARISON CONSTANT
7979 053656 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
7980 053664 016037 000012 001170      MOV      RMS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7981 053672 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
7982 053700 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
7983 053706 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7984 053714 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
7985 053722 016037 000012 001172      MOV      RMS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7986 053730 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
7987 053736 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7988 053744 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7989 053752 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7990 053760 001006      BNE      74$ ;BR IF NOT
7991 053762 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7992 053766 001037      BNE      76$ ;BR IF NOT
7993 053770 104046      ERROR   46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7994 053772 000137 054156      JMP     78$ ;BYPASS THE REST OF THE CHECKS
7995 053776 013737 001170 001126 74$:   MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7996 054004 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7997 054012 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
7998 054020 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7999 054024 001414      BEQ     75$ ;BR IF ZERO
8000 054026 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8001 054034 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8002 054042 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
8003 054050 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8004 054054 001004      BNE      76$ ;BR IF NOT
8005 054056 012737 177777 001254 75$:   MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR

```

F12

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 148
T41 PORT 'B' SET VOLUME VALID TEST

SEQ 0148

8006	054064	104022				ERROR	22	;TYPE ERROR MESSAGE 22
8007	054066	013737	001170	001126	76\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8008	054074	013737	001224	001240		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
8009	054102	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
8010	054110	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
8011	054116	001401				BEQ	77\$;BR IF OK FROM PORT A.
8012	054120	104007				ERROR	7	;REPORT ERROR
8013	054122	013737	001172	001126	77\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8014	054130	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
8015	054136	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
8016	054144	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;SEE IF READ OK FROM PORT B.
8017	054152	001401				BEQ	78\$;BR IF OK
8018	054154	104007				ERROR	7	;REPORT ERROR
8019	054156	000240			78\$:	NOP		
8020	054160	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
8021	054166	013737	001224	001240		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOLT
8022	054174	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
8023	054200	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
8024	054206	012737	000012	001122		MOV	#RMDS1,\$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
8025	054214	060037	001122			ADD	RO,\$BDAOR	;ADD RH11 BASE ADDRESS
8026	054220	012737	100000	001124		MOV	#ATA,\$GDDAT	;WHAT REGISTER SHOULD BE
8027	054226	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
8028	054234	042737	077777	001164		BIC	#!CAT,\$TMP0	;SAVE SPECIFIED BITS
8029	054242	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
8030	054250	001414				BEQ	79\$;BR IF OK
8031	054252	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
8032	054260	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
8033	054266	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
8034	054274	104064				ERROR	64	;TYPE MESSAGE 64
8035	054276	005137	001250			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
8036	054302	000240			79\$:	NOP		
8037	054304	005037	001250			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
8038	054310	016037	000012	001126		MOV	RMDS1(RO), \$BDDAT	;GET CONTENTS OF RMDS1
8039	054316	012737	000012	001122		MOV	#RMDS1,\$BDAOR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
8040	054324	060037	001122			ADD	RO,\$BDAOR	;ADD RH11 BASE ADDRESS
8041	054330	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
8042	054334	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
8043	054342	042737	177677	001164		BIC	#!CVV,\$TMP0	;SAVE SPECIFIED BITS
8044	054350	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
8045	054356	001414				BEQ	81\$;BR IF OK
8046	054360	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
8047	054366	042737	000100	001174		BIC	#VV,\$TMP4	;CLEAR THE MASKED BITS
8048	054374	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
8049	054402	104065				ERROR	65	;TYPE MESSAGE 65
8050	054404	005137	001250			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
8051	054410	000240			81\$:	NOP		
8052								
8053								;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A.
8054								;THEN RELEASE PORT A
8055	054412	012760	000011	000000		MOV	#11, RMCS1(RO)	;DRIVE CLEAR
8056	054420	012760	000021	000000		MOV	#21, RMCS1(RO)	;READ IN PRESET
8057								
8058								;RELEASE THE DRIVE FROM PORT A
8059								
8060	054426	113760	001224	000010		MOVB	PORTA, RMCS2(RO)	;SELECT PORT A
8061	054434	013737	001224	001240		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

8062 054442 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
8063 ;
8064 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8065
8066 054450 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8067 054454 012737 000012 001122 MOV #RMDS1, $BDDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
8068 054462 060037 001122 ADD RO, $BDDADR ;ADD THE I/O BASE ADDRESS
8069 054466 012737 011600 001124 MOV #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
8070 054474 113760 001224 000010 MOVVB PORTA, RMCS2(RO) ;SELECT PORT A.
8071 054502 016037 000012 001170 MOV RMDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8072 054510 042737 024001 001170 BIC #PIP!WAL!OM, $TMP2 ;CLEAR DONT CARES
8073 054516 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
8074 054524 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8075 054532 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
8076 054540 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8077 054546 042737 024001 001172 BIC #PIP!WAL!OM, $TMP3 ;CLEAR DONT CARES
8078 054554 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
8079 054562 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8080 054570 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8081 054576 001006 BNE 83$ ;BR IF NOT
8082 054600 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8083 054604 001037 BNE 85$ ;BR IF NOT
8084 054606 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8085 054610 000137 054774 JMP 87$ ;BYPASS THE REST OF THE CHECKS
8086 054614 013737 001170 001126 83$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8087 054622 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8088 054630 113760 001226 000010 MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
8089 054636 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8090 054642 001414 BEQ 84$ ;BR IF ZERO
8091 054644 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8092 054652 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8093 054660 113760 001224 000010 MOVVB PORTA, RMCS2(RO) ;SELECT PORT A.
8094 054666 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8095 054672 001004 BNE 85$ ;BR IF NOT
8096 054674 012737 177777 001254 84$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
8097 054702 104022 EPR09 22 ;TYPE ERROR MESSAGE 22
8098 054704 013737 001170 001126 85$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8099 054712 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
8100 054720 042737 100100 001126 BIC #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
8101 054726 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
8102 054734 001401 BEQ 86$ ;BR IF OK FROM PORT A.
8103 054736 104007 ERROR 7 ;REPORT ERROR
8104 054740 013737 001172 001126 86$: MOV $TMP3, $BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8105 054746 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
8106 054754 042737 100100 001126 BIC #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
8107 054762 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
8108 054770 001401 BEQ 87$ ;BR IF OK
8109 054772 104007 ERROR 7 ;REPORT ERROR
8110 054774 000240 87$: NOP
8111
8112 054776 000004 88$: SCOPE
8113
8114
8115
8116
8117

```

```

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

```

0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173

```
;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
;*
;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
;*
;* B. WRITE 1'S INTO RMER1 THROUGH PORT 'A' TO FORCE AN ATTENTION.
;*
;* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
;* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND NOT SET FOR
;* PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
;*****
```

```
055000
055000 005737 001300
055004 001406
055006 100002
055010 000137 002676
055014 012737 177777 001300
055022 112737 000042 001102
055030 012737 055052 001106
055036 012737 055052 001110
055044 012737 000002 001176
055052 012706 001100

055056 113760 001224 000010
055064 005060 000012
055070 012760 000011 000000
055076 012760 000013 000000
055104 113760 001226 000010
055112 005060 000012
055116 012760 000011 000000
055124 012760 000013 000000

055132 113760 001224 000010
055140 013737 001224 001242
055146 005060 000012
055152 013737 001226 001244

055160 012760 177777 000014

055166 005037 001256
055172 012737 003720 001260
055200 113760 001226 000010
055206 013737 001226 001242
```

```
†ST42:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #42,$TSTNM ;TEST NUMBER
MOV #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #2,$TIMES ;DO 2 ITERATIONS
TEST42: MOV #STACK,$P ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOV PORTA,RMCS2(R0) ;SELECT PORT #A
CLR RMDS1(R0) ;SEIZE THE DRIVE
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
CLR RMDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
;*****

;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDS1(R0) ;WRITE RMDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
;*****

;FORCE AN ATTENTION BY SETTING ERRORS.
MOV #-1,RMER1(RU) ;SET ERROR BITS
;*****

;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #2000,WATCH ;SET WATCH TO 2000 MS
MOVB PORTB,RMCS2(R0) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
;*****
```



```

8174                                     ;WAIT FOR DRIVE TO TIMEOUT
8175
8176 055214 005760 000012 1$: TST RMDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
8177 055220 001004 BNE 2$ ;BR IF DRIVE RELEASED
8178 055222 005737 001260 TST WATCH ;WATCH AT ZERO ?
8179 055226 001372 BNE 1$ ;BR IF NOT
8180 055230 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
8181 055232
8182 055232 113760 001224 000010 2$: MOVB PORTA, RMC52(RO) ;SELECT PORT A
8183 055240 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8184
8185
8186 :*****
8187 ;THE ERROR BIT ('ERR') IN RMDS1 SHOULD STILL BE SET
8188
8189 055246 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8190 055252 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
8191 055260 012737 000012 001122 MOV #RMDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8192 055266 060037 001122 ADD RO, $BDAOR ;ADD RHI1 BASE ADDRESS
8193 055272 012737 040000 001124 MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
8194 055300 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
8195 055314 042737 137777 001164 BIC #1CERR, $TMP0 ;SAVE SPECIFIED BITS
8196 055322 001414 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
8197 055324 013737 001126 001174 BEQ 66$ ;BR IF OK
8198 055332 042737 040000 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
8199 055340 053737 001174 001124 BIC #ERR, $TMP4 ;CLEAR THE MASKED BITS
8200 055346 104023 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
8201 055350 005137 001250 ERROR 23 ;TYPE MESSAGE 23
8202 055354 000240 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8203 66$: NOP
8204
8205 :*****
8206 ;THE ERROR REGISTER SHOULD CONTAIN 1'S
8207
8208 055356 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8209 055362 016037 000014 001126 MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
8210 055370 012737 000014 001122 MOV #RMER1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8211 055376 060037 001122 ADD RO, $BDAOR ;ADD RHI1 BASE ADDRESS
8212 055402 012737 177777 001124 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
8213 055410 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
8214 055416 001403 BEQ 68$ ;BR IF OK
8215 055420 104010 ERROR 10 ;REPORT THE ERROR
8216 055422 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8217 68$: NOP
8218
8219 :*****
8220 ;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET
8221
8222 055430 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8223 055434 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
8224 055442 012737 000012 001122 MOV #RMDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8225 055450 060037 001122 ADD RO, $BDAOR ;ADD RHI1 BASE ADDRESS
8226 055454 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
8227 055462 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
8228 055470 042737 077777 001164 BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
8229 055476 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
8230 055504 001414 BEQ 70$ ;BR IF OK

```



```

8230 055506 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
8231 055514 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
8232 055522 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
8233 055530 104041          ERROR 41                ;TYPE MESSAGE 41
8234 055532 005137 001250          COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8235 055536 000240          70$:  NOP
8236
8237
8238 ;*****
8239
8240 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8241
8242 055540 005037 001254          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
8243 055544 012737 000012 001122      MOV      #RMS1,$BDAOR    ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
8244 055552 060037 001122          ADD      RO,$BDAOR      ;ADD THE I/O BASE ADDRESS
8245 055556 012737 051700 001124      MOV      #51700,$GDDAT  ;COMPARISON CONSTANT
8246 055564 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
8247 055572 016037 000012 001170      MOV      RMS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8248 055600 042737 024001 001170      BIC      #PIP:WRL:OM,$TMP2 ;CLEAR DONT CARES
8249 055606 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO 'TMP0'
8250 055614 042737 100100 001164      BIC      #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8251 055622 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
8252 055630 016037 000012 001172      MOV      RMS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8253 055636 042737 024001 001172      BIC      #PIP:WRL:OM,$TMP3 ;CLEAR DONT CARES
8254 055644 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO 'TMP1'
8255 055652 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8256 055660 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8257 055666 001006          BNE     72$            ;BR IF NOT
8258 055670 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8259 055674 001045          BNE     74$            ;BR IF NOT
8260 055676 104046          ERROR 46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8261 055700 000137 056100          JMP     76$            ;BYPASS THE REST OF THE CHECKS
8262 055704 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8263 055712 013737 001226 001240          MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8264 055720 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
8265 055726 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
8266 055732 001414          BEQ     73$            ;BR IF ZERO
8267 055734 013737 001224 001240          MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8268 055742 013737 001172 001126          MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
8269 055750 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
8270 055756 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
8271 055762 001012          BNE     74$            ;BR IF NOT
8272 055764 012737 177777 001254 73$:  MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
8273 055772 012760 000011 000000      MOV      #11,RMCS1(RO)  ;CLEAR THE DRIVE
8274 056000 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
8275 056006 104026          ERROR 26              ;TYPE ERROR MESSAGE 26
8276 056010 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8277 056016 013737 001224 001240          MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
8278 056024 042737 100000 001126          BIC      #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
8279 056032 023737 001124 001126          CMP      $GDDAT,$BDDAT  ;ALL BITS OK ?
8280 056040 001401          BEQ     75$            ;BR IF OK FROM PORT A.
8281 056042 104007          ERROR 7               ;REPORT ERROR
8282 056044 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT    ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
8283 056052 013737 001226 001240          MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
8284 056060 042737 100000 001126          BIC      #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
8285 056066 023737 001124 001126          CMP      $GDDAT,$BDDAT  ;SEE IF READ OK FROM PORT B.

```

```

8286 056074 001401
8287 056076 104007
8288 056100 000240
8289
8290
8291
8292
8293 056102 113760 001226 000010
8294 056110 013737 001226 001240
8295 056116 005037 001250
8296 056122 016037 000012 001126
8297 056130 012737 000012 001122
8298 056136 060037 001122
8299 056142 005037 001124
8300 056146 013737 001126 001164
8301 056154 042737 077777 001164
8302 056162 023737 001124 001164
8303 056170 001414
8304 056172 013737 001126 001174
8305 056200 042737 100000 001174
8306 056206 053737 001174 001124
8307 056214 104052
8308 056216 005137 001250
8309 056222 000240
8310
8311
8312
8313 056224 113760 001224 000010
8314 056232 005060 000012
8315 056236 012760 000011 000000
8316 056244 012760 000013 000000
8317 056252 000004
8318
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333 056254
8334 056254 005737 001300
8335 056260 001406
8336 056262 100002
8337 056264 000137 002676
8338 056270 012737 177777 001300
8339 056276 112737 000043 001102
8340 056304 012737 056326 001106
8341 056312 012737 056326 001110

```

```

      BEQ      76$      ;BR IF OK
      ERROR   7        ;REPORT ERROR
76$:  NOP

;*****
;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

      MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
      MOV     PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
      MOV     RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
      MOV     #RMDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      ADD     RO, $B0ADR ;ADD RH11 BASE ADDRESS
      CLR     $GDDAT ;WHAT REGISTER SHOULD BE
      MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
      BIC     #1CATA, $TMP0 ;SAVE SPECIFIED BITS
      CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
      BEQ     77$      ;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

      MOVVB   PORTA, RMCS2(RO) ;SELECT PORT #A
      CLR     RMDS1(RO) ;SEIZE THE DRIVE
      MOV     #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
      MOV     #13, RMCS1(RO) ;RELEASE THE DRIVE
3$:   SCOPE ;LOOP ?

;*****
;TEST 43 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
;
;VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
;
; A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
;
; B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
;
; C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
; NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
; PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
;
;*****
↑ST43:
      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$:   MOV     #-1, KYBCTL ;SET SINGLE TEST INDICATOR
2$:   MOVVB   #43, $STNM ;TEST NUMBER
      MOV     #TEST43, $LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV     #TEST43, $LPERR ;LOAD LOOP ON ERROR ADDRESS

```

```

8342 056320 012737 000002 001176      MOV      #2,$TIMES      ;;DO 2. ITERATIONS
8343 056326 012706 001100      TEST43: MOV      #STACK,SP      ;LOAD THE STACK POINTER
8344
8345      ;CLEAR ATTENTION BITS FOR BOTH PORTS
8346
8347 056332 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT #A
8348 056340 005060 000012 000000      CLR      RMDS1(RO)      ;SEIZE THE DRIVE
8349 056344 012760 000011 000000      MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
8350 056352 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
8351 056360 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT #B
8352 056366 005060 000012 000000      CLR      RMDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
8353 056372 012760 000011 000000      MOV      #11,RMCS1(RO)  ;ISSUE DRIVE CLEAR
8354 056400 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
8355      ;:*****
8356      ;SEIZE THE DRIVE THROUGH PORT B
8357
8358
8359 056406 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
8360 056414 013737 001226 001242      MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
8361 056422 005060 000012 000000      CLR      RMDS1(RO)      ;WRITE RMDS1
8362 056426 013737 001224 001244      MOV      PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
8363
8364      ;:*****
8365      ;FORCE AN ATTENTION BY SETTING ERRORS.
8366
8367 056434 012760 177777 000014      MOV      #-1,RMER1(RO)  ;SET ERROR BITS
8368
8369      ;:*****
8370      ;START THE TIMER
8371
8372 056442 005037 001256      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
8373 056446 012737 003720 001260      MOV      #2000,WATCH   ;SET WATCH TO 2000 MS
8374 056454 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
8375 056462 013737 001224 001240      MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8376
8377      ;:*****
8378      ;WAIT FOR DRIVE TO TIMEOUT
8379
8380 056470 005760 000012      1$: TST      RMDS1(RO)    ;WAIT FOR THE DRIVE TO BE RELEASED
8381 056474 001004      BNE      2$           ;BR IF DRIVE RELEASED
8382 056476 005737 001260      TST      WATCH        ;WATCH AT ZERO ?
8383 056502 001372      BNE      1$           ;BR IF NOT
8384 056504 104036      ERROR   36           ;DRIVE NOT RELEASED WITHIN 2 SECONDS
8385 056506
8386 056506 113760 001226 000010      2$: MOV      PORTB,RMCS2(RO) ;SELECT PORT B
8387 056514 013737 001226 001240      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8388
8389      ;:*****
8390      ;THE ERROR BIT ('ERR') IN RMDS1 SHOULD STILL BE SET
8391
8392 056522 005037 001250      CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
8393 056526 016037 000012 001126      MOV      RMDS1(RO),%BDDAT ;GET CONTENTS OF RMDS1
8394 056534 012737 000012 001122      MOV      #RMDS1,%BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8395 056542 060037 001122      ADD      RO,%BDAOR    ;ADD RHI1 BASE ADDRESS
8396 056546 012737 040000 001124      MOV      #ERR,%GDAT   ;WHAT REGISTER SHOULD BE
8397 056554 013737 001126 001164      MOV      %BDDAT,%TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'

```

```

8398 056562 042737 137777 001164 BIC #ICERR,$TMP0 ;SAVE SPECIFIED BITS
8399 056570 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
8400 056576 001414 BEQ 66$ ;BR IF OK
8401 056600 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
8402 056606 042737 040000 001174 BIC #ERR,$TMP4 ;CLEAR THE MASKED BITS
8403 056614 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
8404 056622 104023 ERROR 23 ;TYPE MESSAGE 23
8405 056624 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8406 056630 000240 66$: NOP

```

;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

8411 056632 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8412 056636 016037 000014 001126 MOV #RMER1($R0),$BDDAT ;GET CONTENTS OF RMER1
8413 056644 012737 000014 001122 MOV #RMER1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8414 056652 060037 001122 ADD $R0,$BDAOR ;ADD RHI1 BASE ADDRESS
8415 056656 012737 177777 001124 MOV #177777,$GDDAT ;WHAT REGISTER SHOULD BE
8416 056664 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK?
8417 056672 001403 BEQ 68$ ;BR IF OK
8418 056674 104010 ERROR 10 ;REPORT THE ERROR
8419 056676 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8420 056702 000240 68$: NOP

```

;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

```

8425 056704 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8426 056710 016037 000012 001126 MOV #RMDS1($R0),$BDDAT ;GET CONTENTS OF RMDS1
8427 056716 012737 000012 001122 MOV #RMDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8428 056724 060037 001122 ADD $R0,$BDAOR ;ADD RHI1 BASE ADDRESS
8429 056730 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
8430 056736 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
8431 056744 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
8432 056752 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
8433 056760 001414 BEQ 70$ ;BR IF OK
8434 056762 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
8435 056770 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
8436 056776 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
8437 057004 104041 ERROR 41 ;TYPE MESSAGE 41
8438 057006 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8439 057012 000240 70$: NOP

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

8446 057014 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8447 057020 012737 000012 001122 MOV #RMDS1,$BDAOR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
8448 057026 060037 001122 ADD $R0,$BDAOR ;ADD THE I/O BASE ADDRESS
8449 057032 012737 051700 001124 MOV #51700,$GDDAT ;COMPARISON CONSTANT
8450 057040 113760 001224 000010 MOVB PORTA,#CMCS2($R0) ;SELECT PORT A
8451 057046 016037 000012 001170 MOV #RMDS1($R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
8452 057054 042737 024001 001170 BIC #PIP:WRL:OM,$TMP2 ;CLEAR DONT CARES
8453 057062 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'

```

```

8454 057070 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8455 057076 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
8456 057104 016037 000012 001172 MOV RMDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8457 057112 042737 024001 001172 BIC #PIPI!WRL!OM,$TMP3 ;CLEAR DONT CARES
8458 057120 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
8459 057126 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8460 057134 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8461 057142 001006 BNE 72$ ;BR IF NOT
8462 057144 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8463 057150 001045 BNE 74$ ;BR IF NOT
8464 057152 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8465 057154 000137 057354 JMP 76$ ;BYPASS THE REST OF THE CHECKS
8466 057160 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8467 057166 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8468 057174 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
8469 057202 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A
8470 057206 001414 BEQ 73$ ;BR IF ZERO
8471 057210 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8472 057216 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8473 057224 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
8474 057232 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8475 057236 001012 BNE 74$ ;BR IF NOT
8476 057240 012737 177777 001254 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8477 057246 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
8478 057254 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
8479 057262 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
8480 057264 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8481 057272 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8482 057300 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
8483 057306 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
8484 057314 001401 BEQ 75$ ;BR IF OK FROM PORT A.
8485 057316 104007 ERROR 7 ;REPORT ERROR
8486 057320 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8487 057326 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8488 057334 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
8489 057342 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8490 057350 001401 BEQ 76$ ;BR IF OK
8491 057352 104007 ERROR 7 ;REPORT ERROR
8492 057354 000240 75$: NOP
8493
8494
8495 :*****
8496 :THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET
8497 057356 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
8498 057364 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8499 057372 005037 001250 CLR #KERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8500 057376 016037 000012 001126 MOV RMDS1(RO), $BDDAT ;GET CONTENTS OF RMDS1
8501 057404 012737 000012 001122 MOV #RMDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8502 057412 060037 001122 ADD RO,$BDDADR ;ADD RMDS1 BASE ADDRESS
8503 057416 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
8504 057422 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
8505 057430 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
8506 057436 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
8507 057444 001414 BEQ 77$ ;BR IF OK
8508 057446 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
8509 057454 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS

```

```

8510 057462 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TIMEOUT
8511 057470 104052 ERROR 52 ;TYPE MESSAGE 52
8512 057472 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8513 057476 000240 77$: NOP
8514 ;CLEAR ATTENTION BIT FOR PORT B
8515
8516
8517 057500 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT #B
8518 057506 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
8519 057512 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
8520 057520 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
8521 057526 000004 3$: SCOPE ;LOOP ?
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565

```

```

*****
*TEST 44 PORT 'A' RETRIGGER BY DEMAND TEST
*
*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
*
* B. WAIT 500 MS AND READ RMDS1 THROUGH PORT 'A'.
*
* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
* TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
*
* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
* BIT IS SET.
*
*****

```

```

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

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

8554 057606 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT #A
8555 057614 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE
8556 057620 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
8557 057626 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
8558 057634 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT #B
8559 057642 005060 000012 CLR RMDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
8560 057646 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
8561 057654 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE

```

;;*****

;SEIZE THE DRIVE THROUGH PORT A

```

8566
8567 057662 113760 001224 000010      MOV      PORTA, RMCS2(RO) ; SELECT PORT A
8568 057670 013737 001224 001242      MOV      PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
8569 057676 005060 000012      CLR      RMDS1(RO) ; WRITE RMDS1
8570 057702 013737 001226 001244      MOV      PORTB, OPPRT ; 'OPPOSITE' PORT ADDRESS
8571
8572 ;*****
8573 ;WAIT 500 MS
8574
8575
8576 ;*****
8577 ;START THE TIMER
8578
8579 057710 005037 001256      CLR      TIME ; CLEAR THE ELAPSED TIME COUNTER
8580 057714 012737 000764 001260      MOV      #500., WATCH ; SET WATCH TO 500 MS
8581 057722 005737 001260      1$: TST     WATCH ; WATCH EQUAL TO ZERO
8582 057726 001375      BNE     1$ ; BR IF NOT
8583
8584 ;*****
8585 ;START THE TIMER
8586
8587 057730 005037 001256      CLR      TIME ; CLEAR THE ELAPSED TIME COUNTER
8588 057734 012737 003720 001260      MOV      #2000., WATCH ; SET WATCH TO 2000 MS
8589
8590 ;*****
8591 ;RETRIGGER THE TIMEOUT ONE-SHOT
8592
8593 057742 005760 000012      TST     RMDS1(RO) ; RETRIGGER THE ONE-SHOT
8594 057746 113760 001226 000010      MOV     PORTB, RMCS2(RO) ; SELECT PORT B
8595 057754 013737 001226 001240      MOV     PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8596 057762 005760 000012      2$: TST     RMDS1(RO) ; WAIT FOR TIMEOUT
8597 057766 001004      BNE     3$ ; BR IF TIMEOUT OCCURRED
8598 057770 005737 001260      TST     WATCH ; WATCH EQUAL TO ZERO ?
8599 057774 001372      BNE     2$ ; BR IF NOT
8600 057776 104036      ERROR  36 ; NO TIMEOUT WITHIN 2 SECONDS
8601 060000 013737 001256 001276      3$: MOV     TIME, TIMES ; SAVE THE ELAPSED TIME VALUE
8602
8603 ;*****
8604
8605 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8606
8607 060006 005037 001254      CLR     RELERR ; CLEAR THE 'RELEASE ERROR' INDICATOR
8608 060012 012737 000012 001122      MOV     #RMDS1, $BDAOR ; FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
8609 060020 060037 001122      ADD     RO, $BDAOR ; ADD THE I/O BASE ADDRESS
8610 060024 012737 011700 001124      MOV     #MOL:PGM:OPR:DRY:VV,$GDDAT ; COMPARISON CONSTANT
8611 060032 113760 001224 000010      MOV     PORTA, RMCS2(RO) ; SELECT PORT A.
8612 060040 016037 000012 001170      MOV     RMDS1(RO), $TMP2 ; GET THE DRIVE STATUS REGISTER FROM PORT A.
8613 060046 042737 024001 001170      BIC     #PIP:WRL:OM, $TMP2 ; CLEAR DONT CARES
8614 060054 013737 001170 001164      MOV     $TMP2, $TMP0 ; COPY IT INTO 'TMP0'
8615 060062 042737 100100 001164      BIC     #ATA:VV, $TMP0 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
8616 060070 113760 001226 000010      MOV     PORTB, RMCS2(RO) ; SELECT PORT B.
8617 060076 016037 000012 001172      MOV     RMDS1(RO), $TMP3 ; GET THE DRIVE STATUS REGISTER FROM PORT B.
8618 060104 042737 024001 001172      BIC     #PIP:WRL:OM, $TMP3 ; CLEAR DONT CARES
8619 060112 013737 001172 001166      MOV     $TMP3, $TMP1 ; COPY IT INTO 'TMP1'
8620 060120 042737 100100 001166      BIC     #ATA:VV, $TMP1 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
8621 060126 023737 001164 001166      CMP     $TMP0, $TMP1 ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```



```

8622 060134 001006 BNE 66$ ;BR IF NOT
8623 060136 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8624 060142 001045 BNE 68$ ;BR IF NOT
8625 060144 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8626 060146 000137 060332 JMP 70$ ;BYPASS THE REST OF THE CHECKS
8627 060152 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8628 060160 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8629 060166 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
8630 060174 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8631 060200 001414 BEQ 67$ ;BR IF ZERO
8632 060202 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8633 060210 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8634 060216 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
8635 060224 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8636 060230 001012 BNE 68$ ;BR IF NOT
8637 060232 012737 177777 001254 67$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
8638 060240 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
8639 060246 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
8640 060254 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
8641 060256 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8642 060264 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8643 060272 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
8644 060300 001401 BEQ 69$ ;BR IF OK FROM PORT A.
8645 060302 104007 ERROR 7 ;REPORT ERROR
8646 060304 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8647 060312 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8648 060320 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8649 060326 001401 BEQ 70$ ;BR IF OK
8650 060330 104007 ERROR 7 ;REPORT ERROR
8651 060332 000240 70$: NOP

```

```

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

```

```

8656 060334 023737 001276 001264 CMP TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
8657 060342 003004 BGT 4$ ;BR IF GREATER
8658 060344 023737 001276 001266 CMP TIMES,TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
8659 060352 002001 BGE +4 ;BR IF NOT
8660 060354 104025 4$: ERROR 25 ;REPORT THE ERROR
8661 060356 000004 SCOPE ;LOOP ?

```

```

*****
*TEST 45 PORT 'B' RETRIGGER BY DEMAND TEST
*
*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
*
* B. WAIT 500 MS AND WRITE 0'B INTO RMDS1 THROUGH PORT 'A'.
*
* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
* TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
*
* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
* BIT IS SET.
*

```

8662
8663
8664
8665
8666
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677

```

8678
8679 060360
8680 060360 005737 001300
8681 060364 001406
8682 060366 100002
8683 060370 000137 002676
8684 060374 012737 177777 001300 1$:
8685 060402 112737 000045 001102 2$:
8686 060410 012737 060432 001106
8687 060416 012737 060432 001110
8688 060424 012737 000002 001176
8689 060432 012706 001100
8690
8691
8692
8693 060436 113760 001224 000010
8694 060444 005060 000012
8695 060450 012760 000011 000000
8696 060456 012760 000013 000000
8697 060464 113760 001226 000010
8698 060472 005060 000012
8699 060476 012760 000011 000000
8700 060504 012760 000013 000000
8701
8702
8703
8704
8705
8706 060512 113760 001226 000010
8707 060520 013737 001226 001242
8708 060526 005060 000012
8709 060532 013737 001224 001244
8710
8711
8712
8713
8714
8715
8716
8717
8718 060540 005037 001256
8719 060544 012737 000764 001260
8720 060552 005737 001260
8721 060556 001375
8722
8723
8724
8725
8726 060560 005037 001256
8727 060564 012737 003720 001260
8728
8729
8730
8731
8732 060572 005760 000012
8733 060576 113760 001224 000010

```

```

;*****
↑ST45:
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 #45,$TSTNM ;TEST NUMBER
MOV #TEST45,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST45,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #2,$TIMES ;DO 2. ITERATIONS
TEST45: MOV #STACK,$P ;LOAD THE STACK POINTER

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

;*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,$RMCS2($RO) ;SELECT PORT B
MOV PORTB,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR $RMS1($RO) ;WRITE $RMS1
MOV PORTA,$OPPRT ;'OPPOSITE' PORT ADDRESS

;*****
;WAIT 500 MS

;*****
;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #500.,WATCH ;SET WATCH TO 500 MS
1$: TST WATCH ;WATCH EQUAL TO ZERO
BNE 1$ ;BR IF NOT

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

;*****
;RETRIGGER THE TIMEOUT ONE-SHOT
TST $RMS1($RO) ;RETRIGGER THE ONE-SHOT
MOVB PORTA,$RMCS2($RO) ;SELECT PORT A

```

```

8734 060604 013737 001224 001240      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8735 060612 005760 000012                TST    RMDS1(RO)   ;WAIT FOR TIMEOUT
8736 060616 001004                BNE    3$         ;BR IF TIMEOUT OCCURRED
8737 060620 005737 001260                TST    WATCH      ;WATCH EQUAL TO ZERO ?
8738 060624 001372                BNE    2$         ;BR IF NOT
8739 060626 104036                ERROR   36        ;NO TIMEOUT WITHIN 2 SECONDS
8740 060630 013737 001256 001276 3$:    MOV    TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
8741
8742 ;*****
8743
8744 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8745
8746 060636 005037 001254                CLR    RELERR     ;CLEAR THE 'RELEASE ERROR' INDICATOR
8747 060642 012737 000012 001122      MOV    #RMDS1,$BDADR ;FORM THE ADDRESS OF RMDS1 FOR TYPEOUT
8748 060650 060037 001122                ADD    RO,$BDADR  ;ADD THE I/O BASE ADDRESS
8749 060654 012737 011700 001124      MOV    #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
8750 060662 113760 001224 000010      MOVVB PORTA, RMCS2(RO) ;SELECT PORT A.
8751 060670 016037 000012 001170      MOV    RMDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8752 060676 042737 024001 001170      BIC    #PIP:WAL:OM,STMP2 ;CLEAR DONT CARES
8753 060704 013737 001170 001164      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
8754 060712 042737 100100 001164      BIC    #ATA:VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8755 060720 113760 001226 000010      MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
8756 060726 016037 000012 001172      MOV    RMDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8757 060734 042737 024001 001172      BIC    #PIP:WAL:OM,STMP3 ;CLEAR DONT CARES
8758 060742 013737 001172 001166      MOV    STMP3,STMP1 ;COPY IT INTO 'STMP1'
8759 060750 042737 100100 001166      BIC    #ATA:VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8760 060756 023737 001164 001166      CMP    STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8761 060764 001006                BNE    66$       ;BR IF NOT
8762 060766 005737 001164                TST    STMP0     ;REGISTERS ARE THE SAME: ARE THEY ZERO ^
8763 060772 001045                BNE    68$       ;BR IF NOT
8764 060774 104046                ERROR   46        ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8765 060776 000137 061162                JMP    70$       ;BYPASS THE REST OF THE CHECKS
8766 061002 013737 001170 001126 66$:    MOV    STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8767 061010 013737 001226 001240      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8768 061016 113760 001226 000010      MOVVB PORTB, RMCS2(RO) ;SELECT PORT B.
8769 061024 005737 001164                TST    STMP0     ;SEE IF STATUS EQ 0 FROM PORT A.
8770 061030 001414                BEQ    67$       ;BR IF ZERO
8771 061032 013737 001224 001240      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8772 061040 013737 001172 001126      MOV    STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8773 061046 113760 001224 000010      MOVVB PORTA, RMCS2(RO) ;SELECT PORT A.
8774 061054 005737 001166                TST    STMP1     ;SEE IF STATUS EQ ZERO FROM PORT B.
8775 061060 001012                BNE    68$       ;BR IF NOT
8776 061062 012737 177777 001254 67$:    MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8777 061070 012760 000011 000000      MOV    #11,RMCS1(RO) ;CLEAR THE DRIVE
8778 061076 012760 000013 000000      MOV    #13,RMCS1(RO) ;RELEASE THE DRIVE
8779 061104 104022                ERROR   22        ;TYPE ERROR MESSAGE 22
8780 061106 013737 001170 001126 68$:    MOV    STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
8781 061114 013737 001224 001240      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
8782 061122 023737 001124 001126      CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
8783 061130 001401                BEQ    69$       ;BR IF OK FROM PORT A.
8784 061132 104007                ERROR   7         ;REPORT ERROR
8785 061134 013737 001172 001126 69$:    MOV    STMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
8786 061142 013737 001226 001240      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
8787 061150 023737 001124 001126      CMP    $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8788 061156 001401                BEQ    70$       ;BR IF OK
8789 061160 104007                ERROR   7         ;REPORT ERROR

```

8790 061162 000240
8791
8792
8793
8794
8795 061164 023737 001276 001272
8796 061172 003004
8797 061174 023737 001276 001274
8798 061202 002001
8799 061204 104025
8800 061206 000004

70\$: NOP
:*****
:CHECK THE TIME FROM RETRIGGER TO TIMEOUT
:*****
CMP TIMES,TIMEBP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
BGT 4\$;BR IF GREATER
CMP TIMES,TIMEBM ;MEASURED TIME LESS THAN -25% TOLERANCE
BGE +4 ;BR IF NOT
4\$: ERROR 25 ;REPORT THE ERROR
SCOPE ;LOOP ?

8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813
8814
8815
8816
8817
8818
8819
8820
8821
8822
8823
8824

:*****
*TEST 46 PORT 'A' TIMEOUT/RELEASE TEST
:*****
*VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
* SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD51.
*
* B. SET PORT REQUEST BY WRITING 0'S INTO RMD51 FROM PORT 'A'.
*
* C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE
* HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
* SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
*
* D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS
* BEEN RELEASED.
*
* E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
:*****

8825 061210
8826 061210 005737 001300
8827 061214 001406
8828 061216 100002
8829 061220 000137 002676
8830 061224 012737 177777 001300
8831 061232 112737 000046 001102
8832 061240 012737 061262 001106
8833 061246 012737 061262 001110
8834 061254 012737 000002 001176
8835 061262 012706 001100

†ST46: 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,\$STSTNM ;TEST NUMBER
MOV #TEST46,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST46,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #2,\$TIMES ;DO 2. ITERATIONS
TEST46: MOV #STACK,\$P ;LOAD THE STACK POINTER

8836
8837
8838
8839 061266 113760 001224 000017
8840 061274 005060 000012
8841 061300 012760 000011 000000
8842 061306 012760 000013 000000
8843 061314 113760 001226 000010
8844 061322 005060 000012
8845 061326 012760 000011 000000

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

```

8846 061334 012760 000013 000000      MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
8847
8848                                     ;:*****
8849
8850                                     ;SEIZE THE DRIVE THROUGH PORT B
8851
8852 061342 113760 001226 000010      MOVB     PORTB, RMCS2(RO) ;SELECT PORT B
8853 061350 013737 001226 001242      MOV      PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
8854 061356 005060 000012                CLR      RMD51(RO) ;WRITE RMD51
8855 061362 013737 001224 001244      MOV      PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
8856 061370 113760 001224 000010      MOVB     PORTA, RMCS2(RO) ;SELECT PORT A
8857 061376 013737 001224 001240      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8858
8859                                     ;:*****
8860                                     ;SET REQUEST THROUGH PORT A
8861
8862 061404 005060 000012                CLR      RMD51(RO) ;SET REQUEST FOR PORT A
8863 061410 113760 001226 000010      MOVB     PORTB, RMCS2(RO) ;SELECT PORT B
8864 061416 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8865
8866                                     ;:*****
8867                                     ;RELEASE THE DRIVE THROUGH PORT B
8868
8869 061424 012760 000013 000000      MOV      #13, RMCS1(RO) ;RELEASE DRIVE THROUGH PORT B
8870
8871                                     ;:*****
8872                                     ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8873
8874 061432 013737 001264 001260      MOV      TIMEAP, WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8875
8876                                     ;:*****
8877                                     ;VERIFY THAT THE DRIVE IS SEIZED BY PORT A
8878
8879 061440 005037 001250                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8880 061444 016037 000012 001126      MOV      RMD51(RO), $BDDAT ;GET CONTENTS OF RMD51
8881 061452 012737 000012 001122      MOV      #RMD51, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8882 061460 060037 001122                ADD      RO, $B0ADR ;ADD RHI1 BASE ADDRESS
8883 061464 005037 001124                CLR      $G0DAT ;WHAT REGISTER SHOULD BE
8884 061470 023737 001124 001126      CMP      $G0DAT, $BDDAT ;IS THE REGISTER OK ?
8885 061476 001403                BEQ      66$ ;BR IF OK
8886 061500 104031                ERROR 31 ;TYPE MESSAGE 31
8887 061502 005137 001250                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8888 061506 000240                NOP ;
8889 061510 005737 001250                TST      CKERR ;REGISTER OK ?
8890 061514 001402                BEQ      +6 ;BR IF OK
8891 061516 000137 062072                JMP      IS ;BYPASS REST OF TEST IF NOT
8892
8893
8894                                     ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
8895 061522 005737 001260                TST      WATCH ;WATCH EQUAL ZERO ?
8896 061526 001375                BNE      -4 ;BR IF NOT
8897
8898                                     ;:*****
8899                                     ;CONFIRM THAT THE DRIVE HAS TIMED OUT
8900
8901

```

:VERIFY THAT THE DRIVE IS IN NEUTRAL

```

8902
8903
8904 061530 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8905 061534 012737 000012 001122 MOV #RMS1,$BDDADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
8906 061542 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
8907 061546 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8908 061554 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A.
8909 061562 016037 000012 001170 MOV RMS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8910 061570 042737 024001 001170 BIC #PIP!WAL!OM,$TMP2 ;CLEAR DONT CARES
8911 061576 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
8912 061604 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8913 061612 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B.
8914 061620 016037 000012 001172 MOV RMS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8915 061626 042737 024001 001172 BIC #PIP!WAL!OM,$TMP3 ;CLEAR DONT CARES
8916 061634 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
8917 061642 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8918 061650 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8919 061656 001006 68$ BNE ;BR IF NOT
8920 061660 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8921 061664 001045 70$ BNE ;BR IF NOT
8922 061666 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8923 061670 000137 062070 JMP 72$ ;BYPASS THE REST OF THE CHECKS
8924 061674 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8925 061702 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8926 061710 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B.
8927 061716 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8928 061722 001414 69$ BEQ ;BR IF ZERO
8929 061724 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8930 061732 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8931 061740 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A.
8932 061746 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8933 061752 001012 70$ BNE ;BR IF NOT
8934 061754 012737 177777 001254 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8935 061762 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
8936 061770 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
8937 061776 104035 ERROR 35 ;TYPE ERROR MESSAGE 35
8938 062000 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
8939 062006 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8940 062014 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
8941 062022 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
8942 062030 001401 71$ BEQ ;BR IF OK FROM PORT A.
8943 062032 104007 ERROR 7 ;REPORT ERROR
8944 062034 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
8945 062042 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8946 062050 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
8947 062056 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
8948 062064 001401 72$ BEQ ;BR IF OK
8949 062066 104007 ERROR 7 ;REPORT ERROR
8950 062070 000240 72$: NOP
8951 062072 000004 1$: SCOPE ;LOOP ?

```

```

8952
8953 *****
8954 *TEST 47 PORT 'B' TIMEOUT/RELEASE TEST
8955 *
8956 *VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
8957 * SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

```

- 8958
- 8959
- 8960
- 8961
- 8962
- 8963
- 8964
- 8965
- 8966
- 8967
- 8968
- 8969
- 8970
- 8971
- 8972
- 8973
- 8974
- 8975
- 8976
- 8977
- 8978
- 8979
- 8980
- 8981
- 8982
- 8983
- 8984
- 8985
- 8986
- 8987
- 8988
- 8989
- 8990
- 8991
- 8992
- 8993
- 8994
- 8995
- 8996
- 8997
- 8998
- 8999
- 9000
- 9001
- 9002
- 9003
- 9004
- 9005
- 9006
- 9007
- 9008
- 9009
- 9010
- 9011
- 9012
- 9013

- * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
- * B. SET PORT REQUEST BY WRITING 0'S INTO RMDS1 FROM PORT 'B'.
- * C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
- * D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

TST47:

```

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$:      MOV         #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOVVB      #47,$TSTNM ;TEST NUMBER
        MOV         #TEST47,$LPADR ;LOAD LOOP ON TEST ADDRESS
        MOV         #TEST47,$LPERR ;LOAD LOOP ON ERROR ADDRESS
        MOV         #2,$TIMES    ;DO 2. ITERATIONS
TEST47:  MOV         #STACK,SP   ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

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

```

;SEIZE THE DRIVE THROUGH PORT A

```

MOVVB    PORTA,RMCS2(RO) ;SELECT PORT A
MOV      PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
CLR      RMDS1(RO)      ;WRITE RMDS1
MOV      PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
MOVVB    PORTB,RMCS2(RO) ;SELECT PORT B
MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

;SET REQUEST THROUGH PORT B

```

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

```

;RELEASE THE DRIVE THROUGH PORT A

K13

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 166
T47 PORT 'B' TIMEOUT/RELEASE TEST

SEQ 0166

```

9014
9015 062310 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE DRIVE THROUGH PORT A
9016
9017      ;*****
9018      ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
9019
9020 062316 013737 001272 001260      MOV      TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
9021
9022      ;*****
9023      ;VERIFY THAT THE DRIVE IS SEIZED BY PORT B
9024
9025 062324 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9026 062330 016037 000012 001126      MOV      RMDS1(RO),SBDDAT ;GET CONTENTS OF RMDS1
9027 062336 012737 000012 001122      MOV      #RMDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9028 062344 060037 001122      ADD      RO,SBDAOR ;ADD RHI1 BASE ADDRESS
9029 062350 005037 001124      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
9030 062354 023737 001124 001126      CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
9031 062362 001403      BEQ      66$ ;BR IF OK
9032 062364 104031      ERROR    31 ;TYPE MESSAGE 31
9033 062366 005137 001250      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9034 062372 000240      66$:    NOP
9035 062374 005737 001250      TST      CKERR ;REGISTER OK ?
9036 062400 001402      BEQ      .+6 ;BR IF OK
9037 062402 000137 062756      JMP      1$ ;BYPASS REST OF TEST IF NOT
9038
9039
9040      ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
9041 062406 005737 001260      TST      WATCH ;WATCH EQUAL ZERO ?
9042 062412 001375      BNE      .-4 ;BR IF NOT
9043
9044      ;*****
9045      ;CONFIRM THAT THE DRIVE HAS TIMED OUT
9046
9047
9048
9049      ;VERIFY THAT THE DRIVE IS IN NEUTRAL
9050 062414 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
9051 062420 012737 000012 001122      MOV      #RMDS1,SBDAOR ;FORM THE ADDRESS OF RMDS1 FOR TIMEOUT
9052 062426 060037 001122      ADD      RO,SBDAOR ;ADD THE I/O BASE ADDRESS
9053 062432 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
9054 062440 113760 001224 000010      MOVB    PORTA,RMCS2(RO) ;SELECT PORT A.
9055 062446 016037 000012 001170      MOV      RMDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
9056 062454 042737 024001 001170      BIC      #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
9057 062462 013737 001170 001164      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
9058 062470 042737 100100 001164      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9059 062476 113760 001226 000010      MOVB    PORTB,RMCS2(RO) ;SELECT PORT B.
9060 062504 016037 000012 001172      MOV      RMDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
9061 062512 042737 024001 001172      BIC      #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
9062 062520 013737 001172 001166      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
9063 062526 042737 100100 001166      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9064 062534 023737 001164 001166      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
9065 062542 001006      BNE      68$ ;BR IF NOT
9066 062544 005737 001164      TST      STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
9067 062550 001045      BNE      70$ ;BR IF NOT
9068 062552 104046      ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
9069 062554 000137 062754      JMP      72$ ;BYPASS THE REST OF THE CHECKS

```

```

9070 062560 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9071 062566 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9072 062574 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
9073 062602 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
9074 062606 001414 BEQ 69$ ;BR IF ZERO
9075 062610 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9076 062616 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
9077 062624 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
9078 062632 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
9079 062636 001012 BNE 70$ ;BR IF NOT
9080 062640 012737 177777 001254 69$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
9081 062646 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
9082 062654 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
9083 062662 104035 ERROR 35 ;TYPE ERROR MESSAGE 35
9084 062664 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS1 READ
9085 062672 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
9086 062700 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
9087 062706 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
9088 062714 001401 BEQ 71$ ;BR IF OK FROM PORT A.
9089 062716 104007 ERROR 7 ;REPORT ERROR
9090 062720 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS1 FOR BIT FAILURES - FROM PORT B.
9091 062726 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
9092 062734 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
9093 062742 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
9094 062750 001401 BEQ 72$ ;BR IF OK
9095 062752 104007 ERROR 7 ;REPORT ERROR
9096 062754 000240 72$: NOP
9097 062756 000004 1$: SCOPE ;LOOP ?

```

9098
9099
9100
9101
9102
9103
9104
9105
9106
9107
9108
9109
9110
9111
9112
9113
9114
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124
9125

062760

```

*****
*TEST 50 PORT 'A' SEIZE ACCESS TEST
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS1.
*
* B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.
*
* C. READ RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT
* 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* D. CLEAR RMER1, RMER2 THROUGH PORT 'A'.
*
* E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT
* PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS
* SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
* SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
*
* G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
TST50:

```

M13

CZRMGB0 RM03/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 168
T50 PORT 'A' SEIZE ACCESS TEST

SEQ 0168

```

9126 062760 005737 001300          TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
9127 062764 001406                   BEQ      25              ;BR IF NOT
9128 062766 100002                   BPL      15              ;BB IF JUST ENTERED TEST
9129 062770 000137 002676          JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
9130 062774 012737 177777 001300 15:   MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
9131 063002 112737 000050 001102 25:   MOVVB   #50,$TSTNM      ;TEST NUMBER
9132 063010 012737 063032 001106   MOV      #TEST50,$LPADR ;LOAD LOOP ON TEST ADDRESS
9133 063016 012737 063032 001110   MOV      #TEST50,$LPERR ;LOAD LOOP ON ERROR ADDRESS
9134 063024 012737 000031 001176   MOV      #25,$TIMES     ;DO 25 ITERATIONS
9135 063032 012706 001100 TEST50: MOV     #STACK,$P      ;LOAD THE STACK POINTER
9136
9137                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
9138
9139 063036 113760 001224 000010   MOVVB   PORTA, RMCS2(RO) ;SELECT PORT #A
9140 063044 005060 000012                   CLR      RMD51(RO)      ;SEIZE THE DRIVE
9141 063050 012760 000011 000000   MOV      #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
9142 063056 012760 000013 000000   MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
9143 063064 113760 001226 000010   MOVVB   PORTB, RMCS2(RO) ;SELECT PORT #B
9144 063072 005060 000012                   CLR      RMD51(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
9145 063076 012760 000011 000000   MOV      #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
9146 063104 012760 000013 000000   MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
9147
9148                                     ;SEIZE THE DRIVE THROUGH PORT A
9149
9150 063112 113760 001224 000010   MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
9151 063120 013737 001224 001242   MOV      PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
9152 063126 005060 000012                   CLR      RMD51(RO)      ;WRITE RMD51
9153 063132 013737 001226 001244   MOV      PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
9154 063140 012760 177777 000014   MOV      #-1, RMER1(RO)  ;LOAD 1'S INTO RMER1 THROUGH PORT A
9155 063146 012760 177777 000042   MOV      #-1, RMER2(RO)  ;LOAD 1'S INTO RMER2 THROUGH PORT A
9156 063154 113760 001226 000010   MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
9157 063162 013737 001226 001240   MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9158 063170 004737 064030          JSR      PC, T50B        ;CHECK THE REGISTERS THROUGH PORT B
9159 063174 113760 001224 000010   MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
9160 063202 013737 001224 001240   MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9161 063210 005060 000042                   CLR      RMER2(RO)      ;CLEAR RMER2 ON PORT A
9162 063214 005060 000014                   CLR      RMER1(RO)      ;CLEAR RMER1 ON PORT A
9163 063220 013760 001236 000016   MOV      ASR1, RMAS(RO)  ;CLEAR THE ATTENTION BIT FOR PORT A
9164 063226 113760 001226 000010   MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
9165 063234 013737 001226 001240   MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9166 063242 012760 177777 000014   MOV      #-1, RMER1(RO)  ;LOAD 1'S INTO RMER1 THROUGH PORT B
9167 063250 012760 177777 000042   MOV      #-1, RMER2(RO)  ;LOAD 1'S INTO RMER2 THROUGH PORT B
9168 063256 113760 001224 000010   MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
9169 063264 013737 001224 001240   MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9170 063272 004737 064030          JSR      PC, T50B        ;CHECK THE REGISTERS THROUGH PORT A
9171
9172                                     ;RELEASE THE DRIVE FROM PORT A
9173
9174 063276 113760 001224 000010   MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
9175 063304 013737 001224 001240   MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9176 063312 012760 000013 000000   MOV      #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
9177
9178                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
9179
9180 063320 005037 001254          CLR      RELERR          ;CLEAR 'RELEASE ERROR INDICATOR
9181 063324 012737 111700 001124   MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT

```

9182	063332	012737	000012	001122	MOV	#RMS1,\$BDAOR	;REGISTER ADDRESS INCREMENT	
9183	063340	060037	001122		ADD	RO,\$BDAOR	;REGISTER BASE ADDRESS FOR TYPEOUT	
9184	063344	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	;SELECT PORT B	
9185	063352	013737	001226	001240	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT	
9186	063360	016037	000012	001164	MOV	RMS1(RO),\$TMP0	;READ STATUS REGISTER FROM PORT B	
9187	063366	113760	001224	000010	MOVB	PORTA,RMCS2(RO)	;SELECT PORT A	
9188	063374	013737	001224	001240	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT	
9189	063402	016037	000012	001126	MOV	RMS1(RO),\$BDDAT	;DRIVE STATUS FROM PORT A	
9190	063410	001404			BEG	66\$;BR IF STATUS FROM PORT A ZERO	
9191	063412	005737	001164		TST	\$TMP0	;IS STATUS FROM PORT B ZERO ?	
9192	063416	001401			BEG	66\$;BR IF ZERO	
9193	063420	104031			ERROR	31	;REPORT DRIVE IN NEUTRAL	
9194	063422	013737	001164	001126	66\$:	MOV	\$TMP0,\$BDDAT	;CHECK STATUS FROM PORT B
9195	063430	013737	001226	001240	MOV	PORTB,PTNBR	;CHANGE PORT ADDRESS FOR TYPEOUT	
9196	063436	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;COMPARE WITH CONSTANT	
9197	063444	001401			BEG	67\$;BR IF OK	
9198	063446	104027			ERROR	27	;REPORT REGISTER ERROR	
9199	063450	000240			67\$:	NOP		
9200								
9201							;RELEASE THE DRIVE FROM PORT B	
9202								
9203	063452	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	;SELECT PORT B	
9204	063460	013737	001226	001240	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT	
9205	063466	012760	000013	000000	MOV	#13,RMCS1(RO)	;ISSUE RELEASE THROUGH PORT B	
9206								
9207							;VERIFY THAT THE DRIVE IS IN NEUTRAL	
9208								
9209	063474	005037	001254		CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR	
9210	063500	012737	000012	001122	MOV	#RMS1,\$BDAOR	;FORM THE ADDRESS OF RMS1 FOR TYPEOUT	
9211	063506	060037	001122		ADD	RO,\$BDAOR	;ADD THE I/O BASE ADDRESS	
9212	063512	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT	
9213	063520	113760	001224	000010	MOVB	PORTA,RMCS2(RO)	;SELECT PORT A	
9214	063526	016037	000012	001170	MOV	RMS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.	
9215	063534	042737	024001	001170	BIC	#PIP!WAL!OM,\$TMP2	;CLEAR DONT CARES	
9216	063542	013737	001170	001164	MOV	\$TMP2,\$TMP0	;COPY IT INTO 'TMP0'	
9217	063550	042737	100100	001164	BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY	
9218	063556	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	;SELECT PORT B.	
9219	063564	016037	000012	001172	MOV	RMS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.	
9220	063572	042737	024001	001172	BIC	#PIP!WAL!OM,\$TMP3	;CLEAR DONT CARES	
9221	063600	013737	001172	001166	MOV	\$TMP3,\$TMP1	;COPY IT INTO 'TMP1'	
9222	063606	042737	100100	001166	BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY	
9223	063614	023737	001164	001166	CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?	
9224	063622	001006			BNE	68\$;BR IF NOT	
9225	063624	005737	001164		TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?	
9226	063630	001045			BNE	70\$;BR IF NOT	
9227	063632	104046			ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED	
9228	063634	000137	064020		JMP	72\$;BYPASS THE REST OF THE CHECKS	
9229	063640	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9230	063646	013737	001226	001240	MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
9231	063654	113760	001226	000010	MOVB	PORTB,RMCS2(RO)	;SELECT PORT B.	
9232	063662	005737	001164		TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.	
9233	063666	001414			BEG	69\$;BR IF ZERO	
9234	063670	013737	001224	001240	MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL	
9235	063676	013737	001172	001126	MOV	\$TMP3,\$BDDAT	;BAD DATA' FOR ERROR TYPE OUT	
9236	063704	113760	001224	000010	MOVB	PORTA,RMCS2(RO)	;SELECT PORT A.	
9237	063712	005737	001166		TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.	

```

9238 063716 001012
9239 063720 012737 177777 001254 69$: BNE 70$ ;BR IF NOT
9240 063726 012760 000011 000000 MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
9241 063734 012760 000013 000000 MOV #11,RMCSI(RO) ;CLEAR THE DRIVE
9242 063742 104026 MOV #13,RMCSI(RO) ;RELEASE THE DRIVE
9243 063744 013737 001170 001126 70$: ERROR 26 ;TYPE ERROR MESSAGE 26
9244 063752 013737 001224 001240 MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMCSI READ
9245 063760 023737 001124 001126 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
9246 063766 001401 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
9247 063770 104007 BEQ 71$ ;BR IF OK FROM PORT A.
9248 063772 013737 001172 001126 71$: ERROR 7 ;REPORT ERROR
9249 064000 013737 001226 001240 MOV $TMP3,$BDDAT ;CHECK RMCSI FOR BIT FAILURES - FROM PORT B
9250 064006 023737 001124 001126 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
9251 064014 001401 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
9252 064016 104007 BEQ 72$ ;BR IF OK
9253 064020 000240 ERROR 7 ;REPORT ERROR
9254 064022 000004 NOP
9255 064024 000137 064252 JMP TST51 ;LOOP ?
9256 ;GO TO THE NEXT TEST
9257 ;CHECK THE REGISTERS ON THE SELECTED PORT
9258
9259 064030 TST50B:
9260 064030 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9261 064034 016037 000014 001126 MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
9262 064042 012737 000014 001122 MOV #RMER1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9263 064050 060037 001122 ADD RO,$BDAOR ;ADD RHI1 BASE ADDRESS
9264 064054 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
9265 064060 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
9266 064066 001403 BEQ 64$ ;BR IF OK
9267 064070 104006 ERROR 6 ;TYPE MESSAGE 6
9268 064072 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9269 064076 016037 000000 001126 64$: MOV RMCSI(RO), $BDDAT ;GET THE CONTENTS OF RMCSI
9270 064104 012737 000000 001122 MOV #RMCSI,$BDAOR ;FORM ADDRESS OF REGISTER
9271 064112 060037 001122 ADD RO,$BDAOR ;ADDRESS BASE
9272 064116 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
9273 064124 001404 BEQ 65$ ;BR IF NOT
9274 064126 104011 ERROR 11 ;REPORT THE ERROR
9275 064130 012760 040000 000000 65$: MOV #TRE,RMCSI(RO) ;CLEAR 'MCPE'
9276 064136 000240 NOP
9277 064140 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9278 064144 016037 000042 001126 MOV RMER2(RO), $BDDAT ;GET CONTENTS OF RMER2
9279 064152 012737 000042 001122 MOV #RMER2,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9280 064160 060037 001122 ADD RO,$BDAOR ;ADD RHI1 BASE ADDRESS
9281 064164 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
9282 064170 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
9283 064176 001403 BEQ 66$ ;BR IF OK
9284 064200 104006 ERROR 6 ;TYPE MESSAGE 6
9285 064202 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9286 064206 016037 000000 001126 66$: MOV RMCSI(RO), $BDDAT ;GET THE CONTENTS OF RMCSI
9287 064214 012737 000000 001122 MOV #RMCSI,$BDAOR ;FORM ADDRESS OF REGISTER
9288 064222 060037 001122 ADD RO,$BDAOR ;ADDRESS BASE
9289 064226 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
9290 064234 001404 BEQ 67$ ;BR IF NOT
9291 064236 104011 ERROR 11 ;REPORT THE ERROR
9292 064240 012760 040000 000000 67$: MOV #TRE,RMCSI(RO) ;CLEAR 'MCPE'
9293 064246 000240 NOP

```

```

9294 064250 000207
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317
9318
9319
9320
9321 064252
9322 064252 005737 001300
9323 064256 001406
9324 064260 100002
9325 064262 000137 002676
9326 064266 012737 177777 001300
9327 064274 112737 000051 001102
9328 064302 012737 064324 001106
9329 064310 012737 064324 001110
9330 064316 012737 000031 001176
9331 064324 012706 001100
9332
9333
9334
9335 064330 113760 001224 000010
9336 064336 005060 000012
9337 064342 012760 000011 000000
9338 064350 012760 000013 000000
9339 064356 113760 001226 000010
9340 064364 005060 000012
9341 064370 012760 000011 000000
9342 064376 012760 000013 000000
9343
9344
9345
9346 064404 113760 001226 000010
9347 064412 013737 001226 001242
9348 064420 005060 000012
9349 064424 013737 001224 001244

```

```

RTS PC ;RETURN
*****
*TEST S1 PORT 'B' SEIZE ACCESS TEST
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS1.
*
* B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.
*
* C. READ RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT
* 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* D. CLEAR RMER1, RMER2 THROUGH PORT 'B'.
*
* E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT
* PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS
* SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
* SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
*
* G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
↑ST51:
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 #51,ST51 ;TEST NUMBER
MOV #TEST51,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST51,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25 ITERATIONS
TEST51: MOV #STACK,SP ;LOAD THE STACK POINTER

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

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

```

```

9350 064432 012760 177777 000014 MOV # -1,RMER1(RO) ;LOAD 1'S INTO RMER1 THROUGH PORT B
9351 064440 012760 177777 000042 MOV # -1,RMER2(RO) ;LOAD 1'S INTO RMER2 THROUGH PORT B
9352 064446 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A
9353 064454 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9354 064462 004737 065322 PC,T51B ;CHECK THE REGISTERS THROUGH PORT A
9355 064466 113760 001226 000010 MOVVB PORTB,RMCS2(RO) ;SELECT PORT B
9356 064474 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9357 064502 005060 000042 CLR RMER2(RO) ;CLEAR RMER2 ON PORT B
9358 064506 005060 000014 CLR RMER1(RO) ;CLEAR RMER1 ON PORT B
9359 064512 013760 001236 000016 MOV ASR1,RMAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT B
9360 064520 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A
9361 064526 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9362 064534 012760 177777 000014 MOV # -1,RMER1(RO) ;LOAD 1'S INTO RMER1 THROUGH PORT A
9363 064542 012760 177777 000042 MOV # -1,RMER2(RO) ;LOAD 1'S INTO RMER2 THROUGH PORT A
9364 064550 113760 001226 000010 MOVVB PORTB,RMCS2(RO) ;SELECT PORT B
9365 064556 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9366 064564 004737 065322 JSR PC,T51B ;CHECK THE REGISTERS THROUGH PORT B
9367
9368 ;RELEASE THE DRIVE FROM PORT B
9369
9370 064570 113760 001226 000010 MOVVB PORTB,RMCS2(RO) ;SELECT PORT B
9371 064576 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9372 064604 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
9373
9374 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
9375
9376 064612 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
9377 064616 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
9378 064624 012737 000012 001122 MOV #RMDS1,$BDAOR ;REGISTER ADDRESS INCREMENT
9379 064632 060037 001122 ADD RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
9380 064636 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A
9381 064644 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9382 064652 016037 000012 001164 MOV RMDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT A
9383 064660 113760 001226 000010 MC/8 PORTB,RMCS2(RO) ;SELECT PORT B
9384 064666 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9385 064674 016037 000012 001126 MOV RMDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
9386 064702 001404 BEQ 66$ ;BR IF STATUS FROM PORT B ZERO
9387 064704 005737 001164 TST $TMPD ;IS STATUS FROM PORT A ZERO ?
9388 064710 001401 BEQ 66$ ;BR IF ZERO
9389 064712 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
9390 064714 013737 001164 001126 66$: MOV $TMPD,$BDDAT ;CHECK STATUS FROM PORT A
9391 064722 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
9392 064730 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
9393 064736 001401 BEQ 67$ ;BR IF OK
9394 064740 104027 ERROR 27 ;REPORT REGISTER ERROR
9395 064742 000240 67$: NOP
9396
9397 ;RELEASE THE DRIVE FROM PORT A
9398
9399 064744 113760 001224 000010 MOVVB PORTA,RMCS2(RO) ;SELECT PORT A
9400 064752 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
9401 064760 012760 000013 000000 MOV #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
9402
9403 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
9404
9405 064766 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR

```



```

9406 064772 012737 000012 001122 MOV #RMS1,$B0ADR ;FORM THE ADDRESS OF RMS1 FOR TYPEOUT
9407 065000 060037 001122 ADD RO,$B0ADR ;ADD THE I/O UNSE ADDRESS
9408 065004 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDAT ;COMPARISON CONSTANT
9409 065012 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
9410 065020 016037 000012 001170 MOV RMS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
9411 065026 042737 024001 001170 BIC #PIP!WAL!OM, $TMP2 ;CLEAR DONT CARES
9412 065034 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
9413 065042 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9414 065050 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9415 065056 016037 000012 001172 MOV RMS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
9416 065064 042737 024001 001172 BIC #PIP!WAL!OM, $TMP3 ;CLEAR DONT CARES
9417 065072 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
9418 065100 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
9419 065106 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
9420 065114 001006 BNE 68$ ;BR IF NOT
9421 065116 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
9422 065122 001045 BNE 70$ ;BR IF NOT
9423 065124 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
9424 065126 000137 065312 JMP 72$ ;BYPASS THE REST OF THE CHECKS
9425 065132 013737 001170 001126 68$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
9426 065140 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9427 065146 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
9428 065154 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
9429 065160 001414 BEQ 69$ ;BR IF ZERO
9430 065162 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
9431 065170 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
9432 065176 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
9433 065204 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
9434 065210 001012 BNE 70$ ;BR IF NOT
9435 065212 012737 177777 001254 69$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
9436 065220 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
9437 065226 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
9438 065234 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
9439 065236 013737 001170 001126 70$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMS1 READ
9440 065244 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
9441 065252 023737 001124 001126 CMP $GDAT, $BDDAT ;ALL BITS OK ?
9442 065260 001401 BEQ 71$ ;BR IF OK FROM PORT A.
9443 065262 104007 ERROR 7 ;REPORT ERROR
9444 065264 013737 001172 001126 71$: MOV $TMP3, $BDDAT ;CHECK RMS1 FOR BIT FAILURES - FROM PORT B.
9445 065272 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
9446 065300 023737 001124 001126 CMP $GDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
9447 065306 001401 BEQ 72$ ;BR IF OK
9448 065310 104007 ERROR 7 ;REPORT ERROR
9449 065312 000240 72$: NOP
9450 065314 000004 SCOPE
9451 065316 000137 065544 JMP TST52 ;LOOP ?
9452 ;GO TO THE NEXT TEST
9453 ;CHECK THE REGISTERS ON THE SELECTED PORT
9454
9455 065322 TST51B:
9456 065322 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9457 065326 016037 000014 001126 MOV RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
9458 065334 012737 000014 001122 MOV #RMER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9459 065342 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
9460 065346 005037 001124 CLR $GDAT ;WHAT REGISTER SHOULD BE
9461 065352 023737 001124 001126 CMP $GDAT, $BDDAT ;IS THE REGISTER OK ?

```



```

9462 065360 001403 BEQ 64$ ;BR IF OK
9463 065362 104006 ERROR 6 ;TYPE MESSAGE 6
9464 065364 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9465 065370 016037 000000 001126 64$: MOV RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RMCS1
9466 065376 012737 000000 001122 MOV #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
9467 065404 060037 001122 ADD RO, $BDADR ;ADDRESS BASE
9468 065410 032737 020000 001126 BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
9469 065416 001404 BEQ 65$ ;BR IF NOT
9470 065420 104011 ERROR 11 ;REPORT THE ERROR
9471 065422 012760 040000 000000 MOV #TRE, RMCS1(RO) ;CLEAR 'MCPE'
9472 065430 000240 65$: NOP
9473 065432 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
9474 065436 016037 000042 001126 MOV RMER2(RO), $BDDAT ;GET CONTENTS OF RMER2
9475 065444 012737 000042 001122 MOV #RMER2, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
9476 065452 060037 001122 ADD RO, $BDADR ;ADD RHI1 BASE ADDRESS
9477 065456 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
9478 065462 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
9479 065470 001403 BEQ 66$ ;BR IF OK
9480 065472 104006 ERROR 6 ;TYPE MESSAGE 6
9481 065474 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
9482 065500 016037 000000 001126 66$: MOV RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RMCS1
9483 065506 012737 000000 001122 MOV #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
9484 065514 060037 001122 ADD RO, $BDADR ;ADDRESS BASE
9485 065520 032737 020000 001126 BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
9486 065526 001404 BEQ 67$ ;BR IF NOT
9487 065530 104011 ERROR 11 ;REPORT THE ERROR
9488 065532 012760 040000 000000 MOV #TRE, RMCS1(RO) ;CLEAR 'MCPE'
9489 065540 000240 67$: NOP
9490 065542 0002U7 RTS PC ;RETURN

```

```

9491
9492
9493 065544 000004 ;*****
↑T52: SCOPE

```

```

9494
9495 .SBTTL END OF PASS ROUTINE

```

```

9496
9497 ;*****
9498 ;*INCREMENT THE PASS NUMBER ($PASS)
9499 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
9500 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
9501 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
9502 ;*IF THERES A MONITOR GO TO IT
9503 ;*IF THERE ISN'T JUMP TO T51AA
9504

```

```

9505 065546 $EOP:
9506 065546 005737 001300 TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
9507 065552 001402 BEQ +6 ;BR IF NOT
9508 065554 000137 002676 JMP EXEC ;RETURN TO KEYBOARD CONTROL
9509 065560 005037 001102 CLR $STNM ;ZERO THE TEST NUMBER
9510 065564 005037 001176 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
9511 065570 005237 001100 INC $PASS ;INCREMENT THE PASS NUMBER
9512 065574 042737 100000 001100 BIC #100000, $PASS ;DON'T ALLOW A NEG. NUMBER
9513 065602 005327 DEC (PC)+ ;LOOP?
9514 065604 000001 $EGPCT: .WORD 1
9515 065606 003063 BGT $DOAGN ;:YES
9516 065610 012737 MOV (PC)+, 2(PC)+ ;:RFSTORE COUNTER
9517 065612 000001 $ENDCT: .WORD 1

```

```

9518 065614 065604 $EOPCT
9519 065616 104401 065624 TYPE 655 ;:TYPE ASCIZ STRING
9520 065622 000407 BR 645 ;:GET OVER THE ASCIZ
9521 ;:655: .ASCIZ <12><15>/END PASS #/
9522 065642 645:
9523 065642 013746 001100 MOV $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
9524 ;:TYPE PASS NUMBER
9525 065646 104405 TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN
9526 065650 104401 065656 TYPE 675 ;:TYPE ASCIZ STRING
9527 065654 000421 BR 665 ;:GET OVER THE ASCIZ
9528 ;:675: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
9529 665:
9530 065720 MOV $ERTTL,-(SP) ;:SAVE $ERTTL FOR TYPEOUT
9531 065720 013746 001112 ;:TOTAL NUMBER OF ERRORS
9532 065724 104405 TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN
9533 065726 104401 001207 TYPE $CRLF ;:TYPE CARRIAGE RETURN, LINE FEED
9534 065732 005037 001112 CLR $ERTTL ;:CLEAR ERROR TOTAL
9535 065736 013700 000042 $GET42: MOV @#42,R0 ;:GET MONITOR ADDRESS
9536 065742 001405 BEQ $DOAGN ;:BRANCH IF NO MONITOR
9537 065744 000005 RESET ;:CLEAR THE WORLD
9538 065746 004710 $ENDAD: JFR PC,(R0) ;:GO TO MONITOR
9539 065750 000240 NOP ;:SAVE ROOM
9540 065752 000240 NOP ;:FOR
9541 065754 000240 NOP ;:ACT11
9542 $DOAGN:
9543 065756 000137 JMP @PC)+ ;:RETURN
9544 065760 003160 $RTNAD: .WORD TST1AA
9545 065762 377 000 $ENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
9546 065766 .EVEN
9547
9548 ;:*****
9549
9550 .SBTTL *** SUBROUTINES ***
9551
9552 ;:*****
9553
9554
9555 ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
9556 ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
9557
9558 065766 012737 066036 000004 CKCLK: MOV #CKCLK1,@ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
9559 065774 005037 000006 CLR @ERRVEC+2 ;NEW PSW
9560 066000 005777 113206 TST @SLKCSR ;CHECK FOR KW11-P
9561 066004 013701 001216 MOV $LPVEC,R1 ;KW11-P VECTOR ADDRESS
9562 066010 012721 066120 MOV #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
9563 066014 012711 000300 MOV #300,(R1) ;PSW - PRI 6
9564 066020 012777 113166 MOV #-1,@SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
9565 066026 012777 000135 113156 MOV #135,@SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
9566 066034 000425 BR CKCLK3
9567 066036 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
9568 066042 012737 066100 000004 MOV #CKCLK2,@ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
9569 066050 005777 113144 TST @SLKS ;LOOK FOR KW11-L
9570 066054 013701 001222 MOV $LLVEC,R1 ;KW11-L VECTOR ADDRESS
9571 066060 012721 066120 MOV #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
9572 066064 012711 000300 MOV #300,(R1) ;PSW - PRI 6
9573 066070 012777 000100 113122 MOV #100,@SLKS ;SET KW11-L INTERRUPT

```

*** SUBROUTINES ***

```

9574 066076 000404
9575 066100 062706 000004
9576 066104 062716 000002
9577 066110 012737 000006 000004 CKCLK3: MOV #5,0ERRVEC ;RESTORE THE ERROR VECTOR
9578 066116 000207 RTS PC
9579
9580 ;ROUTINE TO COUNT CLOCK TICKS
9581
9582 066120 062737 000021 001256 CLOCK: ADD #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
9583 066126 103003 1$ BCC 1$ ;BRANCH IF NO OVERFLOW
9584 066130 012737 177777 001256 1$: MOV #1,TIME ;OVERFLOW - RESTORE MAXIMUM COUNT
9585 066136 005737 001260 1$: TST WATCH ;IS WATCH ALREADY ZERO?
9586 066142 001406 2$ BEQ 2$ ;BR IF IT IS
9587 066144 162737 000021 001260 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
9588 066152 100002 2$ BPL 2$ ;BR IF NOT MINUS
9589 066154 005037 001260 CLR WATCH ;CLEAR WATCH DOG COUNTER
9590 066160 000002 2$: RTI ;RETURN
9591
9592 ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
9593
9594 066162 162706 000004 TOLER: SUB #4,SP ;SETUP STACK
9595 066166 016616 000004 MOV 4(SP), (SP) ;SAVE STACK
9596 066172 013546 MOV @R5+-(SP) ;GET TIME VALUE
9597 066174 011666 000004 MOV (SP),4(SP) ;MOVE TIME VALUE
9598 066200 011666 000006 MOV (SP),6(SP) ;MOVE VALUE AGAIN
9599 066204 006216 ASR (SP) ;DIVIDE BY 2
9600 066206 006216 ASR (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
9601 066210 061666 000004 ADD (SP),4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
9602 066214 162666 000004 SUB (SP)+,4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
9603 06622C 000205 RTS R5 ;RETURN WITH TOLERANCES ON THE STACK
9604
9605 ::*****
9606
9607 .SBTTL 'SYSMAC' UTILITY ROUTINES
9608
9609 ::*****
9610
9611 .SBTTL SCOPE HANDLER ROUTINE
9612
9613 ::*****
9614 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9615 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY:7:0)
9616 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY(15:08)
9617 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9618 ;*SW14=1 LOOP ON TEST
9619 ;*SW11=1 INHIBIT ITERATIONS
9620 ;*SW09=1 LOOP ON ERROR
9621 ;*CALL SCOPE ;:SCOPE=IOT
9622 ;*
9623
9624 066222 $SCOPE:
9625 066222 104407 CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
9626 066224 032777 040000 112706 1$: BIT #BIT14,0SWR ;:LOOP ON PRESENT TEST
9627 066232 001101 BNE $OVER ;:YES IF SW14=1
9628 ;*****START OF CODE FOR THE XOR TESTER*****
9629 066234 000416 $XTSTR: BR 6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE

```

```

9630
9631 066236 013746 000004          MOV      Q#ERRVEC, -(SP)      ; THIS INSTRUCTION TO A "NOP" (NOP=240)
9632 066242 012737 066262 000004  MOV      #55, Q#ERRVEC      ; SAVE THE CONTENTS OF THE ERROR VECTOR
9633 066250 005737 177060          TST      Q#177060          ; SET FOR TIMEOUT
9634 066254 012637 000004          MOV      (SP)+, Q#ERRVEC    ; TIME OUT ON XOR?
9635 066260 000453          BR       $SVLAD            ; RESTORE THE ERROR VECTOR
9636 066262 022626          5$: CMP   (SP)+, (SP)+      ; GO TO THE NEXT TEST
9637 066264 012637 000004          MOV      (SP)+, Q#ERRVEC    ; CLEAR THE STACK AFTER A TIME OUT
9638 066270 000413          BR       7$                ; RESTORE THE ERROR VECTOR
9639 066272          6$: : *****END OF CODE FOR THE XOR TESTER***** ; LOOP ON THE PRESENT TEST
9640 066272 105737 001103          2$: TSTB  $ERFLG            ; HAS AN ERROR OCCURRED?
9641 066276 001421          BEQ     3$                  ; BR IF NO
9642 066300 123737 001115 001103          CMPB   $ERMAX, $ERFLG      ; MAX. ERRORS FOR THIS TEST OCCURRED?
9643 066306 101015          BHI    3$                    ; BR IF NO
9644 066310 032777 001000 112622          BIT    #BIT09, QSWR        ; LOOP ON ERROR?
9645 066316 001404          BEQ    4$                    ; BR IF NO
9646 066320 013737 001110 001106 7$: MOV   $LPERR, $LPADR      ; SET LOOP ADDRESS TO LAST SCOPE
9647 066326 000443          BR     $OVER                ;
9648 066330 105037 001103          4$: CLRB $ERFLG            ; ZERO THE ERROR FLAG
9649 066334 005037 001176          CLR   $TIMES                ; CLEAR THE NUMBER OF ITERATIONS TO MAKE
9650 066340 000415          BR     1$                    ; ESCAPE TO THE NEXT TEST
9651 066342 032777 004000 112570 3$: BIT  #BIT11, QSWR        ; INHIBIT ITERATIONS?
9652 066350 001011          BNE   1$                    ; BR IF YES
9653 066352 005737 001100          TST   $PASS                 ; IF FIRST PASS OF PROGRAM
9654 066356 001406          BEQ   1$                    ; INHIBIT ITERATIONS
9655 066360 005237 001104          INC   $ICNT                 ; INCREMENT ITERATION COUNT
9656 066364 023737 001176 001104          CMP   $TIMES, $ICNT        ; CHECK THE NUMBER OF ITERATIONS MADE
9657 066372 002021          BGE   $OVER                ; BR IF MORE ITERATION REQUIRED
9658 066374 012737 000001 001104 1$: MOV  #1, $ICNT            ; REINITIALIZE THE ITERATION COUNTER
9659 066402 013737 066452 001176          MOV   $MXCNT, $TIMES       ; SET NUMBER OF ITERATIONS TO DO
9660 066410 105237 001102          $SVLAD: INCB  $TSTNM        ; COUNT TEST NUMBERS
9661 066414 011637 001106          MOV   (SP), $LPADR         ; SAVE SCOPE LOOP ADDRESS
9662 066420 011637 001110          MOV   (SP), $LPERR        ; SAVE ERROR LOOP ADDRESS
9663 066424 005037 001200          CLR   $ESCAPE              ; CLEAR THE ESCAPE FROM ERROR ADDRESS
9664 066430 112737 000001 001115          MOVB  #1, $ERMAX           ; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9665 066436 013777 001102 112476 $OVER: MOV  $TSTNM, QDISPLAY ; DISPLAY TEST NUMBER
9666 066444 013716 001106          MOV   $LPADR, (SP)        ; FUDGE RETURN ADDRESS
9667 066450 000002          RTI                        ; FIXES PS
9668 066452 000004          $MXCNT: 4                  ; MAX. NUMBER OF ITERATIONS
9669
9670 .SBTTL  ERROR HANDLER ROUTINE
9671
9672 ; *****
9673 ; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
9674 ; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9675 ; *AND GO TO $ERRTYP ON ERROR
9676 ; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9677 ; *SW15=1      HALT ON ERROR
9678 ; *SW13=1      INHIBIT ERROR TYPEOUTS
9679 ; *SW10=1      BELL ON ERROR
9680 ; *CALL
9681 ; *      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9682
9683 $ERROR:
9684          CKSWR
9685          MOVB  $TSTNM, TSTNUM ;:TEST FOR CHANGE IN SOFT-SWR
9686          INCB  $ERFLG        ;:SET THE ERROR FLAG

```

```

9686 066470 001775          BEQ      7$          ;; DON'T LET THE FLAG GO TO ZERO
9687 066472 013777 001102 112442  MOV     $STNM, $DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
9688 066500 032777 002000 112432  BIT     #BIT10, $SWR    ;; BELL ON ERROR?
9689 066506 001402          BEQ      1$          ;; NO - SKIP
9690 066510 104401 001202          TYPE    $BELL        ;; RING BELL
9691 066514 005237 001112 1$:      INC     $ERTTL       ;; COUNT THE NUMBER OF ERRORS
9692 066520 011637 001116          MOV     (SP), $ERRPC  ;; GET ADDRESS OF ERROR INSTRUCTION
9693 066524 162737 000002 001116  SUB     #2, $ERRPC
9694 066532 117737 112360 001114  MOVB   $ERRPC, $ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
9695 066540 032777 020000 112372  BIT     #BIT13, $SWR  ;; SKIP TYPEOUT IF SET
9696 066546 001004          BNE     20$        ;; SKIP TYPEOUTS
9697 066550 004737 066606          JSR     PC, $ERRTYP  ;; GO TO USER ERROR ROUTINE
9698 066554 104401 001207          TYPE    , $CRLF
9699 066560          20$:
9700 066560 005777 112354 2$:      TST     $SWR          ;; HALT ON ERROR
9701 066564 100002          BPL     3$          ;; SKIP IF CONTINUE
9702 066566 000000          HALT
9703 066570 104407          CKSWR
9704 066572          3$:
9705 066572 022737 065746 000042  CMP     #SENDAD, $#42 ;; ACT-11 AUTO-ACCEPT?
9706 066600 001001          BNE     6$          ;; BRANCH IF NO
9707 066602 000000          HALT
9708 066604          6$:
9709 066604 000002          RTI          ;; RETURN
9710          .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
9711
9712          ;; *****
9713          ;; *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9714          ;; *ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
9715          ;; *AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
9716
9717 066606          $ERRTYP:
9718 066606 104401 001207          TYPE    $CRLF        ;; "CARRIAGE RETURN" & "LINE FEED"
9719 066612 010046          MOV     RO, -(SP)    ;; SAVE RO
9720 066614 005000          CLR     RO          ;; PICKUP THE ITEM INDEX
9721 066616 153700 001114          BISB   $#ITEMB, RO
9722 066622 001004          BNE     1$          ;; IF ITEM NUMBER IS ZERO, JUST
9723          ;; TYPE THE PC OF THE ERROR
9724 066624 013746 001116          MOV     $ERRPC, -(SP) ;; SAVE $ERRPC FOR TYPEOUT
9725          ;; ERROR ADDRESS
9726 066630 104402          TYPOC
9727 066632 000445          BR     10$        ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9728 066634 005300 1$:      DEC     RO          ;; GET OUT
9729 066636 006300          ASL    RO          ;; ADJUST THE INDEX SO THAT IT WILL
9730 066640 006300          ASL    RO          ;; WORK FOR THE ERROR TABLE
9731 066642 006300          ASL    RO
9732 066644 062700 001310          ADD     # $ERRTB, RO ;; FORM TABLE POINTER
9733 066650 012037 066660          MOV     (RO)+, 2$
9734 066654 001404          BEQ     3$          ;; PICKUP "ERROR MESSAGE" POINTER
9735 066656 104401          TYPE    "ERROR MESSAGE" ;; TYPE THE "ERROR MESSAGE"
9736 066660 000000 2$:      .WORD  0          ;; "ERROR MESSAGE" POINTER GOES HERE
9737 066662 104401 001207          TYPE    $CRLF       ;; "CARRIAGE RETURN" & "LINE FEED"
9738 066666 012037 066676 3$:      MOV     (RO)+, 4$
9739 066672 001404          BEQ     5$          ;; PICKUP "DATA HEADER" POINTER
9740 066674 104401          TYPE    "DATA HEADER" ;; TYPE THE "DATA HEADER"
9741 066676 000000 4$:      .WORD  0          ;; "DATA HEADER" POINTER GOES HERE

```

9742 066700 104401 001207
 9743 066704 010146
 9744 066706 012001
 9745 066710 001415
 9746 066712 012000
 9747 066714 105720
 9748 066716 001003
 9749 066720 013146
 9750 066722 104402
 9751 066724 000402
 9752 066726
 9753 066726 013146
 9754 066730 104405
 9755 066732 005711
 9756 066734 001403
 9757 066736 104401 066756
 9758 066742 000764
 9759
 9760 066744 012601
 9761 066746 012600
 9762 066750 104401 001207
 9763 066754 000207
 9764 066756 020040 000
 9765 066762
 9766
 9767
 9768
 9769
 9770
 9771
 9772
 9773
 9774
 9775
 9776
 9777
 9778
 9779
 9780
 9781
 9782
 9783 066762 105737 001157
 9784 066766 100002
 9785 066770 000000
 9786 066772 000407
 9787 066774 010046
 9788 066776 017600 000002
 9789 067002 112046
 9790 067004 001005
 9791 067006 005726
 9792 067010 012600
 9793 067012 062716 000002
 9794 067016 000002
 9795 067020 122716 000011
 9796 067024 001430
 9797 067026 122716 000200

```

TYPE $CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
5$: MOV R1, -(SP) ;:SAVE R1
   MOV (R0)+, R1 ;:PICKUP "DATA TABLE" POINTER
   BEQ 9$ ;:BR IF NO DATA TO BE TYPED
6$: MOV (R0)+, R0 ;:PICKUP "DATA FORMAT" POINTER
   TSTB (R0)+ ;:"OCTAL" OR "DECIMAL"
   BNE 7$ ;:BR IF DECIMAL
   MOV 2(R1)+, -(SP) ;:SAVE 2(R1)+ FOR TYPEOUT
   TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
   BR 8$
7$: MOV 2(R1)+, -(SP) ;:SAVE 2(R1)+ FOR TYPEOUT
   TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN
8$: TST (R1) ;:IS THERE ANOTHER NUMBER?
   BEQ 9$ ;:BR IF NO
   TYPE 11$ ;:TYPE TWO(2) SPACES
   BR 6$ ;:LOOP
9$: MOV (SP)+, R1 ;:RESTORE R1
10$: MOV (SP)+, R0 ;:RESTORE R0
   TYPE $CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
   RTS PC ;:RETURN
11$: .ASCIZ / / ;:TWO(2) SPACES
   .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 $TPFLG ;:IS THERE A TERMINAL?
      BPL 1$ ;:BR IF YES
      HALT ;:HALT HERE IF NO TERMINAL
      BR 3$ ;:LEAVE
1$: MOV R0, -(SP) ;:SAVE R0
   MOV 22(SP), R0 ;:GET ADDRESS OF ASCIZ STRING
2$: MOV 2(R0)+, -(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$: MOV (SP)+, R0 ;:RESTORE R0
3$: ADD #2, (SP) ;:ADJUST RETURN PC
   RTI ;:RETURN
4$: CMPB #HT, (SP) ;:BRANCH IF <HT>
   BEQ 8$ ;:BRANCH IF NOT <CRLF>
   CMPB #CRLF, (SP) ;:BRANCH IF NOT <CRLF>

```

TYPE ROUTINE

```

9798 067032 001006          BNE      5$
9799 067034 005726          TST      (SP)+          ;; POP <CR><LF> EQUIV
9800 067036 104401          TYPE
9801 067040 001207          $CRLF          ;; TYPE A CR AND LF
9802 067042 105037 067176  CLRB     $CHARCNT      ;; CLEAR CHARACTER COUNT
9803 067046 000755          BR       2$            ;; GET NEXT CHARACTER
9804 067050 004737 067132  5$:     JSR      PC,$TYPEC  ;; GO TYPE THIS CHARACTER
9805 067054 123726 001156  6$:     CMPB     $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
9806 067060 001350          BNE      2$            ;; IF NO GO GET NEXT CHAR.
9807 067062 013746 001154  MOV      $NULL,-(SP)    ;; GET # OF FILLER CHARS. NEEDED
9808                                     ;; AND THE NULL CHAR.
9809 067066 105366 000001  7$:     DECB     1(SP)      ;; DOES A NULL NEED TO BE TYPED?
9810 067072 002770          BLT      6$            ;; BR IF NO--GO POP THE NULL OFF OF STACK
9811 067074 004737 067132  JSR      PC,$TYPEC
9812 067100 105337 067176  DECB     $CHARCNT
9813 067104 000770          BR       7$            ;; DO NOT COUNT AS A COUNT
9814                                     ;; LOOP

```

;HORIZONTAL TAB PROCESSOR

```

9815
9816
9817 067106 112716 000040  8$:     MOVB     #' (SP)          ;; REPLACE TAB WITH SPACE
9818 067112 004737 067132  9$:     JSR      PC,$TYPEC      ;; TYPE A SPACE
9819 067116 132737 000007 067176  BITB     #7,$CHARCNT      ;; BRANCH IF NOT AT
9820 067124 001372          BNE      9$            ;; TAB STOP
9821 067126 005726          TST      (SP)+          ;; POP SPACE OFF STACK
9822 067130 000724          BR       2$            ;; GET NEXT CHARACTER
9823 067132 105777 112012  $TYPEC: TSTB     @STPS          ;; WAIT UNTIL PRINTER IS READY
9824 067136 100375          BPL     $TYPEC
9825 067140 116677 000002 112004  MOVB     2(SP),@STPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
9826 067146 122766 000013 000002  CMPB     #CR,2(SP)        ;; IS CHARACTER A CARRIAGE RETURN?
9827 067154 001003          BNE      1$            ;; BRANCH IF NO
9828 067156 105037 067176  CLRB     $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
9829 067162 000406          BR       $TYPEX
9830 067164 122766 000012 000002  1$:     CMPB     #LF,2(SP)    ;; IS CHARACTER A LINE FEED?
9831 067172 001402          BEQ     $TYPEX          ;; BRANCH IF YES
9832 067174 105227          INCB     (PC)+          ;; COUNT THE CHARACTER
9833 067176 000000          $CHARCNT: WORD 0        ;; CHARACTER COUNT STORAGE
9834 067200 000207          $TYPEX:  RTS      PC

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

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

```

```
9854      *      TYPON      ; ;CALL FOR TYPEOUT
9855      *
9856      *$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
9857      *CALL:
9858      *      MOV      NUM,-(SP)      ; ;NUMBER TO BE TYPED
9859      *      TYPOC      ; ;CALL FOR TYPEOUT
9860
9861 067202 017646 000000      $TYPOS: MOV      2(SP),-(SP)      ; ;PICKUP THE MODE
9862 067206 116637 000001 067425  MOVB     1(SP),%SOFILL      ; ;LOAD ZERO FILL SWITCH
9863 067214 112637 067427      MOVB     (SP)+,%SOMODE+1      ; ;NUMBER OF DIGITS TO TYPE
9864 067220 062716 000002      ADD      #2,(SP)      ; ;ADJUST RETURN ADDRESS
9865 067224 000406      BR      $TYPON
9866 067226 112737 000001 067425  $TYPOC: MOVB     #1,%SOFILL      ; ;SET THE ZERO FILL SWITCH
9867 067234 112737 000006 067427      MOVB     #6,%SOMODE+1      ; ;SET FOR SIX(6) DIGITS
9868 067242 112737 000005 067424  $TYPON: MOVB     #5,%SOCNT      ; ;SET THE ITERATION COUNT
9869 067250 010346      MOV      R3,-(SP)      ; ;SAVE R3
9870 067252 010446      MOV      R4,-(SP)      ; ;SAVE R4
9871 067254 010546      MOV      R5,-(SP)      ; ;SAVE R5
9872 067256 113704 067427      MOVB     %SOMODE+1,R4      ; ;GET THE NUMBER OF DIGITS TO TYPE
9873 067262 005404      NEG      R4
9874 067264 062704 000006      ADD      #6,R4      ; ;SUBTRACT IT FOR MAX. ALLOWED
9875 067270 110437 067426      MOVB     R4,%SOMODE      ; ;SAVE IT FOR USE
9876 067274 113704 067425      MOVB     %SOFILL,R4      ; ;GET THE ZERO FILL SWITCH
9877 067300 016605 000012      MOV      12(SP),R5      ; ;PICKUP THE INPUT NUMBER
9878 067304 005003      CLR      R3      ; ;CLEAR THE OUTPUT WORD
9879 067306 006105      1$: ROL      R5      ; ;ROTATE MSB INTO "C"
9880 067310 000404      BR      3$      ; ;GO DO MSB
9881 067312 006105      2$: ROL      R5      ; ;FORM THIS DIGIT
9882 067314 006105      ROL      R5
9883 067316 006105      ROL      R5
9884 067320 010503      MOV      R5,R3
9885 067322 006103      3$: ROL      R3      ; ;GET LSB OF THIS DIGIT
9886 067324 105337 067426      DECB     %SOMODE      ; ;TYPE THIS DIGIT?
9887 067330 100016      BPL      7$      ; ;BR IF NO
9888 067332 042703 177770      BIC      #177770,R3      ; ;GET RID OF JUNK
9889 067336 001002      BNE      4$      ; ;TEST FOR 0
9890 067340 005709      TST      R4      ; ;SUPPRESS THIS 0?
9891 067342 001403      BEQ      5$      ; ;BR IF YES
9892 067344 005204      4$: INC      R4      ; ;DON'T SUPPRESS ANYMORE 0'S
9893 067346 052703 000060      BIS      #'0,R3      ; ;MAKE THIS DIGIT ASCII
9894 067352 052703 000040      5$: BIS      #' ',R3      ; ;MAKE ASCII IF NOT ALREADY
9895 067356 110337 067422      MOVB     R3,%S$      ; ;SAVE FOR TYPING
9896 067362 104401 067422      TYPE     #S$      ; ;GO TYPE THIS DIGIT
9897 067366 105337 067424      7$: DECB     %SOCNT      ; ;COUNT BY 1
9898 067372 003347      BGT      2$      ; ;BR IF MORE TO DO
9899 067374 002402      BLT      6$      ; ;BR IF DONE
9900 067376 005204      INC      R4      ; ;INSURE LAST DIGIT ISN'T A BLANK
9901 067400 000744      BR      2$      ; ;GO DO THE LAST DIGIT
9902 067402 012605      6$: MOV      (SP)+,R5      ; ;RESTORE R5
9903 067404 012604      MOV      (SP)+,R4      ; ;RESTORE R4
9904 067406 012603      MOV      (SP)+,R3      ; ;RESTORE R3
9905 067410 016666 000002 000004      MOV      2(SP),4(SP)      ; ;SET THE STACK FOR RETURNING
9906 067416 012616      MOV      (SP)+,(SP)
9907 067420 000002      RTI
9908 067422 000      8$: .BYTE 0      ; ;RETURN
9909 067423 000      .BYTE 0      ; ;STORAGE FOR ASCII DIGIT
          ; ;TERMINATOR FOR TYPE ROUTINE
```


9910 067424 000
 9911 067425 000
 9912 067426 000000
 9913
 9914
 9915
 9916
 9917
 9918
 9919
 9920
 9921
 9922
 9923
 9924
 9925 067430
 9926 067430 010046
 9927 067432 010146
 9928 067434 010246
 9929 067436 010346
 9930 067440 010546
 9931 067442 012746 020200
 9932 067446 016605 000020
 9933 067452 100004
 9934 067454 005405
 9935 067456 112766 000055 000001
 9936 067464 005000
 9937 067466 012703 067644 1\$:
 9938 067472 112723 000040
 9939 067476 005002 2\$:
 9940 067500 016001 067634
 9941 067504 160105 3\$:
 9942 067506 002402
 9943 067510 005202
 9944 067512 000774
 9945 067514 060105 4\$:
 9946 067516 005702
 9947 067520 001002
 9948 067522 105716
 9949 067524 100407
 9950 067526 106316 5\$:
 9951 067530 103003
 9952 067532 116663 000001 177777
 9953 067540 052702 000060 6\$:
 9954 067544 052702 000040 7\$:
 9955 067550 110223
 9956 067552 005720
 9957 067554 020027 000010
 9958 067560 002746
 9959 067562 003002
 9960 067564 010502
 9961 067566 000764
 9962 067570 105726 8\$:
 9963 067572 100003
 9964 067574 116663 177777 177776
 9965 067602 105013 9\$:

```

SOCNT: .BYTE 0          ;; OCTAL DIGIT COUNTER
SOFILL: .BYTE 0        ;; ZERO FILL SWITCH
SOMCODE: .WORD 0       ;; NUMBER OF DIGITS TO TYPE
.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

$TYPDS:
MOV      R0,-(SP)          ;; PUSH R0 ON STACK
MOV      R1,-(SP)          ;; PUSH R1 ON STACK
MOV      R2,-(SP)          ;; PUSH R2 ON STACK
MOV      R3,-(SP)          ;; PUSH R3 ON STACK
MOV      R5,-(SP)          ;; PUSH R5 ON STACK
MOV      #20200,-(SP)      ;; SET BLANK SWITCH AND SIGN
MOV      R0(R2),R5         ;; GET THE INPUT NUMBER
BPL      1$                ;; BR IF INPUT IS POS.
NEG      R5                ;; MAKE THE BINARY NUMBER POS.
MOV      #'-',1(SP)        ;; MAKE THE ASCII NUMBER NEG.
CLR      R0                ;; ZERO THE CONSTANTS INDEX
MOV      #D0BLK,R3         ;; SETUP THE OUTPUT POINTER
MOV      #'',(R3)+         ;; SET THE FIRST CHARACTER TO A BLANK
CLR      R2                ;; CLEAR THE BCD NUMBER
MOV      #D0TBL(R0),R1     ;; GET THE CONSTANT
SUB      R1,R5             ;; FORM THIS BCD DIGIT
BLT      4$                ;; BR IF DONE
INC      R2                ;; INCREASE THE BCD DIGIT BY 1
BR       3$

4$:      ADD      R1,R5     ;; ADD BACK THE CONSTANT
TST      R2                ;; CHECK IF BCD DIGIT=0
BNE      5$                ;; FALL THROUGH IF 0
TSTB    (SP)               ;; STILL DOING LEADING 0'S?
BMI      7$                ;; BR IF YES
ASLB    (SP)               ;; MSD?
BCC      6$                ;; BR IF NO
MOV      1(SP),-1(R3)      ;; YES--SET THE SIGN
BIS      #'0,R2           ;; MAKE THE BCD DIGIT ASCII
BIS      #' ,R2           ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOV      R2,(R3)+         ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST      (R0)+            ;; JUST INCREMENTING
CMP      R0,#10           ;; CHECK THE TABLE INDEX
BLT      2$                ;; GO DO THE NEXT DIGIT
BGT      8$                ;; GO TO EXIT
MOV      R5,R2            ;; GET THE LSD
BR       6$                ;; GO CHANGE TO ASCII
TSTB    (SP)+            ;; WAS THE LSD THE FIRST NON-ZERO?
BPL      9$                ;; BR IF NO
MOV      -1(SP),-2(R3)    ;; YES--SET THE SIGN FOR TYPING
CLRB    (R3)              ;; SET THE TERMINATOR

```

```

9966 067604 012605      MOV      (SP)+,R5      ;; POP STACK INTO R5
9967 067606 012603      MOV      (SP)+,R3      ;; POP STACK INTO R3
9968 067610 012602      MOV      (SP)+,R2      ;; POP STACK INTO R2
9969 067612 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
9970 067614 012600      MOV      (SP)+,R0      ;; POP STACK INTO R0
9971 067616 104401 067644    TYPE     $DBLK         ;; NOW TYPE THE NUMBER
9972 067622 016666 000002 000004    MOV      2(SP),4(SP)   ;; ADJUST THE STACK
9973 067630 012616      MOV      (SP)+,SP      ;;
9974 067632 000002      RTI                          ;; RETURN TO USER
9975 067634 023420      SDBL:   10000.
9976 067636 001750      1000.
9977 067640 000144      100.
9978 067642 000012      10.
9979 067644 000001      SDBLK:  .BLKW 4
          .SBTTL TTY INPUT ROUTINE

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

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

; *TK SERVICE ROUTINE
; *THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
; *BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
; *IT IN THE QUEUE.
$TKSRV: MOVB     $TKB,-(SP)   ;; PICKUP THE CHARACTER
        BIC     #177,(SP)    ;; STRIP THE JUNK
1$:     CMP     (SP),#7      ;; IS IT A CONTROL G?
        BNE     2$          ;; BRANCH IF NO
        CMP     #SWREG,SWR   ;; IS SOFT-SWR SELECTED?
        BEQ     6$          ;; GO TO SWR CHANGE
2$:     CMP     #1,$TKCNT    ;; IS THE QUEUE FULL?

```

```

10022 067770 001004      BNE      3$          ;; BRANCH IF NO
10023 067772 104401 001202  TYPE     $BELL      ;; RING THE TTY BELL
10024 067776 005726      TST     (SP)+       ;; CLEAN CHARACTER OFF OF STACK
10025 070000 000451      BR      5$          ;; EXIT
10026 070002 021627 000023 3$:      CMP     (SP),#23   ;; IS IT A CONTROL-S?
10027 070006 001021      BNE     32$        ;; BRANCH IF NO
10028 070010 005077 111130  CLR     2$TKS     ;; DISABLE TTY KEYBOARD INTERRUPTS
10029 070014 005726      TST     (SP)+       ;; CLEAN CHAR OFF STACK
10030 070016 105777 111122 31$:    TSTB    2$TKS     ;; WAIT FOR A CHAR
10031 070022 100375      BPL     31$        ;; LOOP UNTIL ITS THERE
10032 070024 117746 111116  MOVB   2$TKB, -(SP) ;; GET THE CHARACTER
10033 070030 042716 177600  BIC     #1C177, (SP) ;; MAKE IT 7-BIT ASCII
10034 070034 022627 000021  CMP     (SP)+, #21  ;; IS IT A CONTROL-Q?
10035 070040 001366      BNE     31$        ;; BRANCH IF NO
10036 070042 012777 000100 111074  MOV     #100, 2$TKS ;; REENABLE TTY KEYBOARD INTERRUPTS
10037 070050 000002      RTI                    ;; RETURN
10038 070052 005237 067654 32$:    INC     $TKCNT     ;; COUNT THIS CHARACTER
10039 070056 021627 000140  CMP     (SP), #140  ;; IS IT UPPER CASE?
10040 070062 002405      BLT                    ;; BRANCH IF YES
10041 070064 021627 000175  CMP     (SP), #175  ;; IS IT A SPECIAL CHAR?
10042 070070 003002      BGT                    ;; BRANCH IF YES
10043 070072 042716 000040  BIC     #40, (SP)   ;; MAKE IT UPPER CASE
10044 070076 112677 177554 4$:    MOVB   (SP)+, 2$TKQIN ;; AND PUT IT IN QUEUE
10045 070102 005237 067656      INC     $TKQIN     ;; UPDATE THE POINTER
10046 070106 023727 067656 067663  CMP     $TKQIN, #2$TKQEND ;; GO OFF THE END?
10047 070114 001003      BNE     5$          ;; BRANCH IF NO
10048 070116 012737 067662 067656  MOV     #2$TKQRT, $TKQIN ;; RESET THE POINTER
10049 070124 000002      RTI                    ;; RETURN

```

```

10050
10051
10052
10053
10054
10055

```

```

*****
;SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
;ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
;SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
;CALL WHEN OPERATING IN TTY INTERRUPT MODE.
10056 070126 022737 000176 001140 $CKSWR: CMP     #SWREG, SWR    ;; IS THE SOFT-SWR SELECTED
10057 070134 001104      BNE     15$        ;; EXIT IF NOT
10058 070136 105777 111002  TSTB   2$TKS     ;; IS A CHAR WAITING?
10059 070142 100101      BPL     15$        ;; IF NOT, EXIT
10060 070144 117746 110776  MOVB   2$TKB, -(SP) ;; YES
10061 070150 042716 177600  BIC     #1C177, (SP) ;; MAKE IT 7-BIT ASCII
10062 070154 021627 000007  CMP     (SP), #7   ;; IS IT A CONTROL-G?
10063 070160 001300      BNE     2$          ;; IF NOT, PUT IT IN THE TTY QUEUE
10064
10065
10066
10067
10068
10069

```

```

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

```

```

10070 070162 123727 001134 000001 6$:    CMPB   $AUTOB, #1   ;; ARE WE RUNNING IN AUTO-MODE?
10071 070170 001674      BEQ     2$          ;; BRANCH IF YES
10072 070172 005726      TST     (SP)+       ;; CLEAR CONTROL-G OFF STACK
10073 070174 004737 067664  JSR     PC, $TKINT  ;; FLUSH THE TTY INPUT QUEUE
10074 070200 005077 110740  CLR     2$TKS     ;; DISABLE TTY KEYBOARD INTERRUPTS
10075 070204 112737 000001 001135  MOVB   #1, $INTAC  ;; SET INTERRUPT MODE INDICATOR
10076
10077 070212 104401 070770      TYPE     , $CNTLG   ;; ECHO THE CONTROL-G (↑G).

```

10078	070216	104401	070775		\$GTSWR: TYPE	\$MSWR	:: TYPE CURRENT CONTENTS
10079	070222	013746	000176		MOV	\$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
10080	070226	104402			TYPC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
10081	070230	104401	071006		TYPE	,\$MNEW	:: PROMPT FOR NEW SWR
10082	070234	005046		19\$:	CLR	-(SP)	:: CLEAR COUNTER
10083	070236	005046			CLR	-(SP)	:: THE NEW SWR
10084	070240	105777	110700	7\$:	TSTB	\$STKS	:: CHAR THERE?
10085	070244	100375			BPL	7\$:: IF NOT TRY AGAIN
10086							
10087	070246	117746	110674		MOVB	\$STKB, -(SP)	:: PICK UP CHAR
10088	070252	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
10089							
10090							
10091							
10092	070256	021627	000025	9\$:	CMP	(SP), #25	:: IS IT A CONTROL-U?
10093	070262	001005			BNE	10\$:: BRANCH IF NOT
10094	070264	104401	070763		TYPE	,\$CNTLU	:: YES, ECHO CONTROL-U (FL)
10095	070270	062706	000006	20\$:	ADD	#6, SP	:: IGNORE PREVIOUS INPUT
10096	070274	000757			BR	19\$:: LET'S TRY IT AGAIN
10097							
10098							
10099	070276	021627	000015	10\$:	CMP	(SP), #15	:: IS IT A <CR>?
10100	070302	001022			BNE	16\$:: BRANCH IF NO
10101	070304	005766	000004		TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
10102	070310	001403			BEQ	11\$:: BRANCH IF YES
10103	070312	016677	000002	110620	MOV	2(SP), \$SWR	:: SAVE NEW SWR
10104	070320	062706	000006		ADD	#6, SP	:: CLEAR UP STACK
10105	070324	104401	001207		TYPE	,\$CRLF	:: ECHO <CR> AND <LF>
10106	070330	123727	001135	000001	CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
10107	070336	001003			BNE	15\$:: BRANCH IF NOT
10108	070340	012777	000100	110576	MOV	#100, \$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
10109	070346	000002			RTI		:: RETURN
10110	070350	004737	067132		JSR	PC, \$TYPEC	:: ECHO CHAR
10111	070354	021627	000060		CMP	(SP), #60	:: CHAR < 0?
10112	070360	002420			BLT	18\$:: BRANCH IF YES
10113	070362	021627	000067		CMP	(SP), #67	:: CHAR > 7?
10114	070366	003015			BGT	18\$:: BRANCH IF YES
10115	070370	042726	000060		BIC	#60, (SP)+	:: STRIP-OFF ASCII
10116	070374	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
10117	070400	001403			BEQ	17\$:: BRANCH IF YES
10118	070402	006316			ASL	(SP)	:: NO, SHIFT PRESENT
10119	070404	006316			ASL	(SP)	:: CHAR OVER TO MAKE
10120	070406	006316			ASL	(SP)	:: ROOM FOR NEW ONE.
10121	070410	005266	000002		INC	2(SP)	:: KEEP COUNT OF CHAR
10122	070414	056616	177776	17\$:	BIS	-2(SP), (SP)	:: SET IN NEW CHAR
10123	070420	000707			BR	7\$:: GET THE NEXT ONE
10124	070422	104401	001206	18\$:	TYPE	,\$QUES	:: TYPE ?<CR><LF>
10125	070426	000720			BR	20\$:: SIMULATE CONTROL-U
10126					.DSABL	LSB	
10127							
10128							
10129							
10130							
10131							
10132							
10133							

 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 *CALL:
 * ROCHR :: GET A CHARACTER FROM THE QUEUE
 * RETURN HERE :: CHARACTER IS ON THE STACK

```

10134      ;*                               ;; WITH PRIORITY BIT STRIPPED OFF
10135      ;*
10136      ;*
10137      070430 011646      SPDCHR: MOV      (SP), -(SP)           ;; PUSH DOWN THE PC AND
10138      070432 016666 000004 000002  MOV      4(SP), 2(SP)       ;; THE PS
10139      070440 005066 000004      CLR      4(SP)           ;; GET READY FOR A CHARACTER
10140      070444 005046      CLR      -(SP)           ;; PUT NEW PS ON STACK
10141      070446 012746 070454      MOV      #64$, -(SP)      ;; PUT NEW PC ON STACK
10142      070452 000002      RTI                    ;; POP NEW PC AND PS
10143      070454      64$:
10144      070454 005737 067654      1$:  TST      $TKCNT           ;; WAIT ON A CHARACTER
10145      070460 001775      BEQ      1$
10146      070462 005337 067654      DEC      $TKCNT           ;; DECREMENT THE COUNTER
10147      070466 117766 177166 000004  MOVB     2($TKQOUT), 4(SP)  ;; GET ONE CHARACTER
10148      070474 005237 067660      INC      $TKQOUT         ;; UPDATE THE POINTER
10149      070500 023727 067660 067663  CMP      $TKQOUT, #($TKQEND) ;; DID IT GO OFF OF THE END?
10150      070506 001003      BNE      2$              ;; BRANCH IF NO
10151      070510 012737 067662 067660  MOV      #($TKQSR), $TKQOUT ;; RESET THE POINTER
10152      070516 000002      RTI                    ;; RETURN
10153      ;*****
10154      ; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10155      ; CALL:
10156      ;*      RDLIN
10157      ;*      RETURN HERE
10158      ;*
10159      ;*
10160      070520 010346      $RDLIN: MOV      R3, -(SP)           ;; SAVE R3
10161      070522 005046      CLR      -(SP)           ;; CLEAR THE RUBOUT KEY
10162      070524 012703 070754      1$:  MOV      #($TTYIN), R3       ;; GET ADDRESS
10163      070530 022703 070763      2$:  CMP      #($TTYIN+7), R3     ;; BUFFER FULL?
10164      070534 101456      BLOS     4$              ;; BR IF YES
10165      070536 104410      RDCHR           ;; GO READ ONE CHARACTER FROM THE TTY
10166      070540 112613      MOVB     (SP)+, (R3)       ;; GET CHARACTER
10167      070542 122713 000177      10$: CMPB     #177, (R3)        ;; IS IT A RUBOUT
10168      070546 001022      BNE      5$              ;; BR IF NO
10169      070550 005716      TST      (SP)             ;; IS THIS THE FIRST RUBOUT?
10170      070552 001007      BNE      6$              ;; BR IF NO
10171      070554 112737 000134 070752  MOVB     #' \, 9$         ;; TYPE A BACK SLASH
10172      070562 104401 070752      TYPE     9$
10173      070566 012716 177777      MOV      #-1, (SP)        ;; SET THE RUBOUT KEY
10174      070572 005303      6$:  DEC      R3              ;; BACKUP BY ONE
10175      070574 020327 070754      CMP      R3, #($TTYIN)    ;; STACK EMPTY?
10176      070600 103434      BLO      4$              ;; BR IF YES
10177      070602 111337 070752      MOVB     (R3), 9$         ;; SETUP TO TYPEOUT THE DELETED CHAR.
10178      070606 104401 070752      TYPE     9$
10179      070612 000746      BR      2$              ;; GO TYPE
10180      070614 005716      5$:  TST      (SP)             ;; RUBOUT KEY SET?
10181      070616 001406      BEQ      7$              ;; BR IF NO
10182      070620 112737 000134 070752  MOVB     #' \, 9$         ;; TYPE A BACK SLASH
10183      070626 104401 070752      TYPE     9$
10184      070632 005016      CLR      (SP)             ;; CLEAR THE RUBOUT KEY
10185      070634 122713 000025      7$:  CMPB     #25, (R3)        ;; IS CHARACTER A CTRL J?
10186      070640 001003      BNE      8$              ;; BR IF NO
10187      070642 104401 070763      TYPE     $CNTLU           ;; TYPE A CONTROL "U"
10188      070646 000726      BR      1$              ;; GO START OVER
10189      070650 122713 000022      8$:  CMPB     #22, (R3)        ;; IS CHARACTER A "↑"?

```

```

10190 070654 001011          BNE      3$          ;; BRANCH IF NO
10191 070656 105013          CLR      (R3)        ;; CLEAR THE CHARACTER
10192 070660 104401          TYPE    ,SCLF        ;; TYPE A "CR" & "LF"
10193 070664 104401 001207  TYPE    $TTYIN       ;; TYPE THE INPUT STRING
10194 070670 000717          BR       2$          ;; GO PICKUP ANOTHER CHARACTER
10195 070672 104401 001206  4$:     TYPE    $QUES       ;; TYPE A '?'
10196 070676 000712          BR       1$          ;; CLEAR THE BUFFER AND LOOP
10197 070700 111337 070752  3$:     MOV      (R3),9$      ;; ECHO THE CHARACTER
10198 070704 104401 070752  TYPE    9$
10199 070710 122723 000015  CMP      #15,(R3)+   ;; CHECK FOR RETURN
10200 070714 001305          SNE      2$          ;; LOOP IF NOT RETURN
10201 070716 105063 177777  CLR      -1(R3)     ;; CLEAR RETURN (THE 15)
10202 070722 104401 001210  TYPE    $LF          ;; TYPE A LINE FEED
10203 070726 005726          TST     (SP)+        ;; CLEAN RUBOUT KEY FROM THE STACK
10204 070730 012603          MOV     (SP)+,R3    ;; RESTORE R3
10205 070732 011646          MOV     (SP)-,(SP)  ;; ADJUST THE STACK AND PUT ADDRESS OF THE
10206 070734 016666 000004 000002  MOV     4(SP),2(SP) ;; FIRST ASCII CHARACTER ON IT
10207 070742 012766 070754 000004  MOV     $TTYIN,4(SP)
10208 070750 000002          RTI
10209 070752 000          9$:     .BYTE    0          ;; STORAGE FOR ASCII CHAR. TO TYPE
10210 070753 000          .BYTE    0          ;; TERMINATOR
10211 070754 000007          $TTYIN: .BLKB    7          ;; RESERVE 7 BYTES FOR TTY INPUT
10212 070763 136 006525 000012  $CNTLU: .ASCIZ  /↑U/<15><12> ;; CONTROL "U"
10213 070770 043536 005015 000  $CNTLG: .ASCIZ  /↑G/<15><12> ;; CONTROL "G"
10214 070775 015 051412 051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
10215 071002 036440 000040
10216 071006 020040 042516 020127  $MNEW:  .ASCIZ  / NEW = /
10217 071014 020075 000
10218 071020
10219
10220
10221
10222
10223
10224
10225
10226
10227
10228
10229
10230
10231
10232
10233 071020 011646          $RDOCT: MOV     (SP)-,(SP) ;; PROVIDE SPACE FOR THE
10234 071022 016666 000004 000002  MOV     4(SP),2(SP) ;; INPUT NUMBER
10235 071030 010046          MOV     R0,-(SP)    ;; PUSH R0 ON STACK
10236 071032 010146          MOV     R1,-(SP)    ;; PUSH R1 ON STACK
10237 071034 010246          MOV     R2,-(SP)    ;; PUSH R2 ON STACK
10238 071036 104411          1$:     ROLIN          ;; READ AN ASCIZ LINE
10239 071040 012600          MOV     (SP)+,R0    ;; GET ADDRESS OF 1ST CHARACTER
10240 071042 010037 071146  MOV     R0,5$       ;; AND SAVE IT
10241 071046 005001          CLR     R1          ;; CLEAR DATA WORD
10242 071050 005002          CLR     R2
10243 071052 112046          2$:     MOV      (R0)+,-(SP) ;; PICKUP THIS CHARACTER
10244 071054 001420          BEQ     3$          ;; IF ZERO GET OUT
10245 071056 122716 000060  CMP      #0,(SP)    ;; MAKE SURE THIS CHARACTER

```

```

.EVEN
.SBTTL READ AN OCTAL NUMBER FROM THE TTY

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

```

```

10246 071062 003026          BGT      4$          ;; IS AN OCTAL DIGIT
10247 071064 122716 000067  CMPB     #'7,(SP)
10248 071070 002423          BLT      4$
10249 071072 006301          ASL     R1          ;; *2
10250 071074 006102          ROL     R2
10251 071076 006301          ASL     R1          ;; *4
10252 071100 006102          ROL     R2
10253 071102 006301          ASL     R1          ;; *8
10254 071104 006102          ROL     R2
10255 071106 042716 177770  BIC     #'C7,(SP)   ;; STRIP THE ASCII JUNK
10256 071112 062601          ADD     (SP)+,R1   ;; ADD IN THIS DIGIT
10257 071114 000756          BR      2$         ;; LOOP
10258 071116 005726          3$: TST     (SP)+   ;; CLEAN TERMINATOR FROM STACK
10259 071120 010166 000012  MOV     R1,12(SP)  ;; SAVE THE RESULT
10260 071124 010237 071156  MOV     R2,$HIOCT
10261 071130 012602          MOV     (SP)+,R2   ;; POP STACK INTO R2
10262 071132 012601          MOV     (SP)+,R1   ;; POP STACK INTO R1
10263 071134 012600          MOV     (SP)+,R0   ;; POP STACK INTO R0
10264 071136 000002          RTI
10265 071140 005726          4$: TST     (SP)+   ;; CLEAN PARTIAL FROM STACK
10266 071142 105010          CLRB   (R0)        ;; SET A TERMINATOR
10267 071144 104401          TYPE   ;; TYPE UP THRU THE BAD CHAR.
10268 071146 000000          5$: WORD   0
10269 071150 104401 001206  TYPE   $QUES       ;; "?" "CR" & "LF"
10270 071154 000730          BR      1$         ;; TRY AGAIN
10271 071156 000000          $HIOCT: WORD 0     ;; HIGH ORDER BITS GO HERE
10272          .SBTTL SAVE AND RESTORE R0-R5 ROUTINES
10273
10274          ;*****
10275          ;SAVE R0-R5
10276          ;CALL:
10277          ; SAVREG
10278          ;UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10279          ;
10280          ;TOP---(+16)
10281          ; +2---(+18)
10282          ; +4---R5
10283          ; +6---R4
10284          ; +8---R3
10285          ;+10---R2
10286          ;+12---R1
10287          ;+14---R0
10288
10289          $SAVREG:
10290          MOV     R0,-(SP)  ;; PUSH R0 ON STACK
10291          MOV     R1,-(SP)  ;; PUSH R1 ON STACK
10292          MOV     R2,-(SP)  ;; PUSH R2 ON STACK
10293          MOV     R3,-(SP)  ;; PUSH R3 ON STACK
10294          MOV     R4,-(SP)  ;; PUSH R4 ON STACK
10295          MOV     R5,-(SP)  ;; PUSH R5 ON STACK
10296          MOV     22(SP),-(SP) ;; SAVE PS OF MAIN FLOW
10297          MOV     22(SP),-(SP) ;; SAVE PC OF MAIN FLOW
10298          MOV     22(SP),-(SP) ;; SAVE PS OF CALL
10299          MOV     22(SP),-(SP) ;; SAVE PC OF CALL
10300
10301          RTI

```

```

10302
10303
10304
10305 071216
10306 071216 012666 000022
10307 071222 012666 000022
10308 071226 012666 000022
10309 071232 012666 000022
10310 071236 012605
10311 071240 012604
10312 071242 012603
10313 071244 012602
10314 071246 012601
10315 071250 012600
10316 071252 000002
10317
10318
10319
10320
10321
10322
10323
10324
10325 071254 010046
10326 071256 016600 000002
10327 071262 005740
10328 071264 111000
10329 071266 006300
10330 071270 016000 071310
10331 071274 000200
10332
10333
10334
10335
10336 071276 011646
10337 071300 016666 000004 000002
10338 071306 000002
10339
10340
10341
10342
10343
10344
10345
10346
10347 071310 071276
10348 071312 066762
10349 071314 067226
10350 071316 067202
10351 071320 067242
10352 071322 067430
10353
10354 071324 070216
10355
10356 071326 070126
10357 071330 070430

```

```

;*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
;GO TO THAT ROUTINE.
$TRAP: 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 $TRPAD(RO),RO ;;INDEX TO TABLE
RTS RO ;;GO TO ROUTINE

;:THIS IS USE TO HANDLE THE "GETPRI" MACRO
$TRAP2: MOV (SP)-,(SP) ;;MOVE THE PC DOWN
MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
RTI ;;RESTORE THE PSW

.SBTTL TRAP TABLE

;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;BY THE "TRAP" INSTRUCTION.
; ROUTINE
;-----
$TRPAD: .WORD $TRAP2
$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

```


10358 071332 070520
 10359 071334 071020
 10360 071336 071160
 10361 071340 071216
 10362
 10363
 10364
 10365
 10366
 10367
 10368
 10369 071342 005015 041412 051132
 10370 071350 043515 030102 005015
 10371 071356 046522 031460 051057
 10372 071364 030115 020062 052504
 10373 071372 046101 041440 047117
 10374 071400 051124 046117 042514
 10375 071406 020122 047514 044507
 10376 071414 020103 042524 052123
 10377 071422 026440 050040 051101
 10378 071430 020124 006461 005012
 10379 071436 000
 10380 071437 015 042412 052116
 10381 071444 051105 042040 044522
 10382 071452 042526 040440 042104
 10383 071460 042522 051523 020072
 10384 071466 000
 10385 071467 111 053116 046101
 10386 071474 042111 040440 042104
 10387 071502 042522 051523 005015
 10388 071510 000
 10389 071511 015 050012 051117
 10390 071516 020124 020101 042101
 10391 071524 051104 051505 020123
 10392 071532 051511 020072 J00
 10393 071537 015 050012 051117
 10394 071544 020124 020102 042101
 10395 071552 051104 051505 020123
 10396 071560 051511 020072 000
 10397 071565 015 051412 051531
 10398 071572 042524 020115 052515
 10399 071600 052123 044040 053101
 10400 071606 020105 046047 020047
 10401 071614 051117 023440 023520
 10402 071622 041440 047514 045503
 10403 071630 005015 000012
 10404 071634 042412 052116 051105
 10405 071642 052040 051505 020124
 10406 071650 035043 000040
 10407 071654 047111 040526 044514
 10408 071662 020104 042524 052123
 10409 071670 047040 046525 042502
 10410 071676 006522 000012
 10411 071702 042440 051122 051117
 10412 071710 006523 000012
 10413 071714 005015 052012 042510

\$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>/CZRMGB0/<CR><LF>
 .ASCIZ @RM03/RM02 DUAL CONTROLLER LOGIC TEST - PART 10<CR><LF><LF>

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

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

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

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

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

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

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

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

ADDRIS: .ASCIZ <CR><LF><LF>THE PRESENT ADDRESS OF THE PH11 (RMCS1) IS:

10414	071722	050040	042522	042523
10415	071730	052116	040440	042104
10416	071736	042522	051523	047440
10417	071744	020106	044124	020105
10418	071752	044122	030461	024040
10419	071760	046522	051503	024461
10420	071766	044440	035123	000040
10421	071774	042412	052116	051105
10422	072002	047040	053505	051040
10423	072010	030510	020061	042101
10424	072016	051104	051505	035123
10425	072024	000040		
10426				
10427				
10428				
10429				
10430				
10431				
10432				
10433	072026	051127	047117	020107
10434	072034	051104	053111	020105
10435	072042	054524	042520	000
10436				
10437	072047	104	044522	042526
10438	072054	047040	052117	047440
10439	072062	020116	044514	042516
10440	072070	000		
10441				
10442	072071	123	051105	040511
10443	072076	020114	052516	041115
10444	072104	051105	051040	040505
10445	072112	020104	044124	047522
10446	072120	043525	020110	040505
10447	072126	044103	050040	051117
10448	072134	020124	047516	020124
10449	072142	044124	020105	040523
10450	072150	042515	000	
10451				
10452	072153	104	044522	042526
10453	072160	047040	052117	051440
10454	072166	044505	042532	020104
10455	072174	054502	050040	051117
10456	072202	000124		
10457				
10458	072204	051127	047117	020107
10459	072212	052123	052101	051525
10460	072220	051440	042505	020116
10461	072226	054502	052040	042510
10462	072234	051440	044505	044532
10463	072242	043516	050040	051117
10464	072250	000124		
10465				

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

;*****

.SBTTL TEST ERROR MESSAGES

;*****

EM1: .ASCIZ /WRONG DRIVE TYPE/

EM2: .ASCIZ /DRIVE NOT ON LINE/

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

EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/

EM5: .ASCIZ WRONG STATUS SEEN BY THE SEIZING PORT

10466	072252	042522	044507	052123	EM6: .ASCIZ REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
10467	072260	051105	041440	047117	
10468	072266	042524	052116	020123	
10469	072274	042527	042522	051440	
10470	072302	042505	020116	054502	
10471	072310	047440	050120	051517	
10472	072316	052111	020105	047520	
10473	072324	052122	026440	042040	
10474	072332	044522	042526	053440	
10475	072340	051501	051440	044505	
10476	072346	042532	000104		
10477					
10478	072352	042522	044507	052123	EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
10479	072360	051105	041440	047117	
10480	072366	042524	052116	020123	
10481	072374	051127	047117	020107	
10482	072402	043101	042524	020122	
10483	072410	042522	042514	051501	
10484	072416	020105	051117	052040	
10485	072424	046511	047505	052125	
10486	072432	000			
10487					
10488	072433	122	043505	051511	EM10: .ASCIZ /REGISTER CONTENTS WRONG/
10489	072440	042524	020122	047503	
10490	072446	052116	047105	051524	
10491	072454	053440	047522	043516	
10492	072462	000			
10493					
10494	072463	103	047117	051124	EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
10495	072470	046117	041040	051525	
10496	072476	050040	051101	052111	
10497	072504	020131	051105	047522	
10498	072512	020122	042522	042101	
10499	072520	047111	020107	047111	
10500	072526	044504	040503	042524	
10501	072534	020104	042522	044507	
10502	072542	052123	051105	000	
10503					
10504	072547	104	044522	042526	EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
10505	072554	047040	052117	051440	
10506	072562	044505	042532	020104	
10507	072570	054502	042040	044522	
10508	072576	042526	041440	042514	
10509	072604	051101	041440	046517	
10510	072612	040515	042116	000	
10511					
10512	072617	122	040505	044504	EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT
10513	072624	020116	051120	051505	
10514	072632	052105	042040	042517	
10515	072640	020123	047516	020124	
10516	072646	042523	020124	047526	
10517	072654	052514	042515	053040	
10518	072662	046101	042111	043040	
10519	072670	051117	052040	042510	
10520	072676	050040	051117	000124	
10521					

10522	072704	047526	052514	042515	EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
10523	072712	053040	046101	042111	
10524	072720	051440	052105	047440	
10525	072726	020116	044124	020105	
10526	072734	051127	047117	020107	
10527	072742	047520	052122	000	
10528					
10529	072747	101	052124	020116	EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
10530	072754	044502	020124	051127	
10531	072762	047117	020107	043101	
10532	072770	042524	020122	044524	
10533	072776	042515	052517	020124	
10534	073004	020055	042522	052521	
10535	073012	051505	020124	047516	
10536	073020	020124	042523	000124	
10537					
10538	073026	052101	047124	041040	EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
10539	073034	052111	053440	047522	
10540	073042	043516	040440	052106	
10541	073050	051105	051040	046105	
10542	073056	040505	042523	026440	
10543	073064	051040	050505	042525	
10544	073072	052123	051440	052105	
10545	073100	000			
10546					
10547	073101	101	052124	020116	EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
10548	073106	044502	020124	051127	
10549	073114	047117	020107	043101	
10550	073122	042524	020122	042522	
10551	073130	042514	051501	020105	
10552	073136	020055	042522	052521	
10553	073144	051505	020124	047516	
10554	073152	020124	042523	000124	
10555					
10556	073160	051104	053111	020105	EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
10557	073166	047516	020124	042523	
10558	073174	055111	042105	053440	
10559	073202	042510	020116	052101	
10560	073210	047124	041040	052111	
10561	073216	043040	051117	050040	
10562	073224	051117	020124	046103	
10563	073232	040505	042522	000104	

10564						
10565	073240	051104	053111	020105	EM21:	.ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
10566	073246	042523	055111	042105		
10567	073254	053440	042510	020116		
10568	073262	042532	047522	053440		
10569	073270	044522	052124	047105		
10570	073276	044440	020116	052101		
10571	073304	047124	041040	052111		
10572	073312	000				
10573						
10574	073313	104	044522	042526	EM22:	.ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
10575	073320	047040	052117	044440		
10576	073326	020116	042516	052125		
10577	073334	040522	020114	043101		
10578	073342	042524	020122	044524		
10579	073350	042515	052517	020124		
10580	073356	020055	042522	052521		
10581	073364	051505	020124	047516		
10582	073372	020124	042523	000124		
10583						
10584	073400	044524	042515	052517	EM23:	.ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
10585	073406	020124	046103	040505		
10586	073414	042522	020104	044124		
10587	073422	020105	051104	053111		
10588	073430	023505	020123	051105		
10589	073436	047522	020122	044502		
10590	073444	000124				
10591						
10592	073446	042522	042514	051501	EM24:	.ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
10593	073454	020105	047503	046515		
10594	073462	047101	020104	042522		
10595	073470	042514	051501	042105		
10596	073476	042040	044522	042526		
10597	073504	053440	052111	020110		
10598	073512	051105	047522	051522		
10599	073520	051440	052105	000		
10600						
10601	073525	124	046511	047505	E. 5:	.ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
10602	073532	052125	047440	042516		
10603	073540	051455	047510	020124		
10604	073546	044504	020104	047516		
10605	073554	020124	042522	051124		
10606	073562	043511	042507	000122		
10607						
10608	073570	051104	053111	020105	EM26:	.ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET.
10609	073576	047516	020124	047111		
10610	073604	047040	052505	051124		
10611	073612	046101	040440	052106		
10612	073620	051105	051040	046105		
10613	073626	040505	042523	026440		
10614	073634	051040	050505	042525		
10615	073642	052123	047040	052117		
10616	073650	051440	052105	000		
10617						
10618	073655	122	043505	051511	EM27:	.ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET.
10619	073662	042524	020122	051127		

10620	073670	047117	020107	043101	
10621	073676	042524	020122	042522	
10622	073704	042514	051501	020105	
10623	073712	044527	044124	051040	
10624	073720	050505	042525	052123	
10625	073726	051440	052105	000	
10626					
10627	073733	104	044522	042526	EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
10628	073740	051440	044505	042532	
10629	073746	020104	054502	051040	
10630	073754	046105	040505	042523	
10631	073762	041440	046517	040515	
10632	073770	042116	044440	051523	
10633	073776	042525	020104	044127	
10634	074004	047105	042040	044522	
10635	074012	042526	044440	020116	
10636	074020	042516	052125	040522	
10637	074026	000114			
10638					
10639	074030	051104	053111	020105	EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
10640	074036	047111	047040	052505	
10641	074044	051124	046101	040440	
10642	074052	052106	051105	051040	
10643	074060	046105	040505	042523	
10644	074066	026440	051040	050505	
10645	074074	042525	052123	051440	
10646	074102	052105	000		
10647					
10648	074105	101	052124	020116	EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
10649	074112	044502	020124	051127	
10650	074120	047117	020107	043101	
10651	074126	042524	020122	042522	
10652	074134	040503	044514	051102	
10653	074142	052101	020105	047503	
10654	074150	046515	047101	000104	
10655					
10656	074156	051104	053111	020105	EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
10657	074164	042522	052524	047122	
10658	074172	042105	052040	020117	
10659	074200	042516	052125	040522	
10660	074206	020114	043111	042040	
10661	074214	044522	042526	041440	
10662	074222	042514	051101	043440	
10663	074230	053111	047105	053440	
10664	074236	044510	042514	042040	
10665	074244	044522	042526	051440	
10666	074252	044505	042532	000104	
10667					
10668	074260	051104	053111	020105	EM34: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
10669	074266	042522	052524	047122	
10670	074274	042105	052040	020117	
10671	074302	042516	052125	040522	
10672	074310	020114	043111	046440	
10673	074316	051501	041123	051525	
10674	074324	044440	044516	020124	
10675	074332	044507	042526	020116	

10676	074340	044127	046111	020105	
10677	074346	051104	053111	020105	
10678	074354	042523	055111	042105	
10679	074362	000			
10680					
10681	074363	124	046511	047505	EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
10682	074370	052125	047440	042516	
10683	074376	051440	047510	020124	
10684	074404	044506	042522	020104	
10685	074412	044527	044124	052517	
10686	074420	020124	042522	044507	
10687	074426	052123	051105	040440	
10688	074434	041503	051505	000123	
10689					
10690	074442	044524	042515	052517	EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
10691	074450	020124	040510	020123	
10692	074456	047516	020124	041517	
10693	074464	052503	051122	042105	
10694	074472	053440	052111	044510	
10695	074500	020116	020062	042523	
10696	074506	047503	042116	000123	
10697					
10698	074514	051104	053111	020105	EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
10699	074522	051511	047040	047117	
10700	074530	042455	044530	052123	
10701	074536	047105	020124	023450	
10702	074544	042516	023504	041040	
10703	074552	052111	051440	052105	
10704	074560	000051			
10705					
10706	074562	052101	047124	041040	EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
10707	074570	052111	043040	051117	
10708	074576	050040	051117	020124	
10709	074604	047516	020124	042522	
10710	074612	042523	020124	042502	
10711	074620	046440	051501	041123	
10712	074626	051525	041440	042514	
10713	074634	051101	000		
10714					
10715	074637	124	046511	047505	EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
10716	074644	052125	041440	042514	
10717	074652	051101	042105	052040	
10718	074660	042510	040440	052124	
10719	074666	047105	044524	047117	
10720	074674	041040	052111	000	
10721					
10722	074701	104	044522	042526	EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
10723	074706	047040	052117	044440	
10724	074714	020116	042516	052125	
10725	074722	040522	020114	051117	
10726	074730	051440	044505	042532	
10727	074736	020104	043101	042524	
10728	074744	020122	052101	047124	
10729	074752	041040	052111	053440	
10730	074760	044522	052124	047105	
10731	074766	000			

10732					
10733	074767	104	044522	042526	EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
10734	074774	044440	020116	042516	
10735	075002	052125	040522	020114	
10736	075010	043101	042524	020122	
10737	075016	052101	042524	052116	
10738	075024	047511	020116	044502	
10739	075032	020124	051127	052111	
10740	075040	042524	000116	.	
10741					
10742	075044	051127	052111	020105	EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
10743	075052	052101	042524	052116	
10744	075060	047511	020116	044502	
10745	075066	020124	044504	020104	
10746	075074	047516	020124	042523	
10747	075102	020124	047520	052122	
10748	075110	051040	050505	042525	
10749	075116	052123	000	.	
10750					
10751	075121	103	047117	051124	EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
10752	075126	046117	042514	020122	
10753	075134	042523	042514	052103	
10754	075142	051440	044527	041524	
10755	075150	020110	047117	042040	
10756	075156	044522	042526	047040	
10757	075164	052117	044440	020116	
10758	075172	040447	041057	000047	
10759					
10760	075200	040503	023516	020124	EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT
10761	075206	041501	042503	051523	
10762	075214	042040	044522	042526	
10763	075222	052040	051110	052517	
10764	075230	044107	042440	052111	
10765	075236	042510	020122	047520	
10766	075244	052122	000		
10767					
10768	075247	101	052124	020116	EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT.
10769	075254	044502	020124	047506	
10770	075262	020122	042523	055111	
10771	075270	047111	020107	047520	
10772	075276	052122	047040	052117	
10773	075304	041440	042514	051101	
10774	075312	042105	041040	020131	
10775	075320	040515	051523	052502	
10776	075326	020123	047111	052111	
10777	075334	000			
10778					
10779	075335	101	052124	020116	EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR/
10780	075342	044502	020124	047506	
10781	075350	020122	050117	047520	
10782	075356	044523	042524	050040	
10783	075364	051117	020124	046103	
10784	075372	040505	042522	020104	
10785	075400	054502	042040	044522	
10786	075406	042526	041440	042514	
10787	075414	051101	000		

10788					
10789	075417	101	052124	020116	EMS1: .ASCIZ /ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
10790	075424	044502	020124	047516	
10791	075432	020124	046103	040505	
10792	075440	042522	020104	054502	
10793	075446	046440	051501	041123	
10794	075454	051523	044440	044516	
10795	075462	026124	042040	044522	
10796	075470	042526	044440	020116	
10797	075476	042516	052125	040522	
10798	075504	000114			
10799					
10800	075506	044124	020105	052101	EMS2: .ASCIZ /THE ATTN BIT SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
10801	075514	047124	041040	052111	
10802	075522	051440	052105	040440	
10803	075530	052106	051105	052040	
10804	075536	046511	047505	052125	
10805	075544	053440	052111	020110	
10806	075552	047516	051040	050505	
10807	075560	042525	052123	023040	
10808	075566	023440	051105	023522	
10809	075574	051440	052105	000	
10810					
10811	075601	103	047101	052047	EMS3: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
10812	075606	051040	040505	020104	
10813	075614	044124	020105	052101	
10814	075622	047124	041040	052111	
10815	075630	043040	047522	020115	
10816	075636	044124	020105	047447	
10817	075644	050120	051517	052111	
10818	075652	023505	050040	051117	
10819	075660	000124			
10820					
10821	075662	042522	042514	051501	EMS4: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
10822	075670	020105	047503	046515	
10823	075676	047101	020104	042522	
10824	075704	047503	047107	055111	
10825	075712	042105	053440	042510	
10826	075720	020116	051511	052523	
10827	075726	042105	041040	020131	
10828	075734	047516	026516	042523	
10829	075742	055111	047111	020107	
10830	075750	047520	052122	000	
10831					
10832	075755	124	046511	047505	EMS5: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
10833	075762	052125	047440	042516	
10834	075770	051455	047510	020124	
10835	075776	051511	046040	051505	
10836	076004	020123	044124	047101	
10837	076012	032440	030060	046440	
10838	076020	000123			
10839					
10840	076022	044122	030461	042040	EMS6: .ASCIZ /RH11 DIDN'T RESPOND TO ADDRESSING/
10841	076030	042111	023516	020124	
10842	076036	042522	050123	047117	
10843	076044	020104	047524	040440	

10844	076052	042104	042522	051523		
10845	076060	047111	000107			
10846						
10847	076064	047520	052122	051040	EM57:	.ASCIZ /PORT REQUEST FLOP(S) WRONG STATE/
10848	076072	050505	042525	052123		
10849	076100	043040	047514	024120		
10850	076106	024523	053440	047522		
10851	076114	043516	051440	040524		
10852	076122	042524	000			
10853						
10854	076125	101	052124	047105	EM60:	.ASCIZ /ATTENTION NOT RESET BY WRITING RMAS/
10855	076132	044524	047117	047040		
10856	076140	052117	051040	051505		
10857	076146	052105	041040	020131		
10858	076154	051127	052111	047111		
10859	076162	020107	046522	051501		
10860	076170	000				
10861						
10862	076171	101	052124	047105	EM61:	.ASCIZ /ATTENTION NOT RESET BY GO/
10863	076176	044524	047117	047040		
10864	076204	052117	051040	051505		
10865	076212	052105	041040	020131		
10866	076220	047507	000			
10867						
10868	076223	101	052124	047105	EM62:	.ASCIZ /ATTENTION RESET BY GO WHEN NOT SEIZED/
10869	076230	044524	047117	051040		
10870	076236	051505	052105	041040		
10871	076244	020131	047507	053440		
10872	076252	042510	020116	047516		
10873	076260	020124	042523	055111		
10874	076266	042105	000			
10875						
10876	076271	104	044522	042526	EM63:	.ASCIZ /DRIVE SEIZED BY UNIT READY CHANGE/
10877	076276	051440	044505	042532		
10878	076304	020104	054502	052440		
10879	076312	044516	020124	042522		
10880	076320	042101	020131	044103		
10881	076326	047101	042507	000		
10882						
10883	076333	101	052124	047105	EM64:	.ASCIZ /ATTENTION NOT SET BY UNIT READY CHANGE/
10884	076340	044524	047117	047040		
10885	076346	052117	051440	052105		
10886	076354	041040	020131	047125		
10887	076362	052111	051040	040505		
10888	076370	054504	041440	040510		
10889	076376	043516	000105			
10890						
10891	076402	047526	052514	042515	EM65:	.ASCIZ /VOLUME VALID NOT RESET BY UNIT READY/
10892	076410	053040	046101	042111		
10893	076416	047040	052117	051040		
10894	076424	051505	052105	041040		
10895	076432	020131	047125	052111		
10896	076440	051040	040505	054504		
10897	076446	000				
10898						
10899						

10956	077116	020043	042440	051122					
10957	077124	050040	020103	050040					
10958	077132	051117	020124	020043					
10959	077140	051040	043505	040440					
10960	077146	051104	041440	047117					
10961	077154	042524	052116	000123					
10962	077162	020040	020040	020040	DH13:	.ASCII /	SEIZE	ERROR<<CR><LF>	
10963	077170	020040	020040	020040					
10964	077176	020040	020040	042523					
10965	077204	055111	020105	020040					
10966	077212	051105	047522	006522					
10967	077220	012							
10968	077221	124	051505	020124		.ASCIZ /TEST #	ERR PC	PORT #	PORT #
10969	077226	020043	042440	051122				REG ADR	CONTENTS/
10970	077234	050040	020103	050040					
10971	077242	051117	020124	020043					
10972	077250	050040	051117	020124					
10973	077256	020043	051040	043505					
10974	077264	040440	051104	041440					
10975	077272	047117	042524	052116					
10976	077300	000123							
10977	077302	020040	020040	020040	DH22:	.ASCII /	RELSNG	SEIZE<<CR><LF>	
10978	077310	020040	020040	020040					
10979	077316	020040	020040	042522					
10980	077324	051514	043516	020040					
10981	077332	042523	055111	006505					
10982	077340	012							
10983	077341	124	051505	020124		.ASCIZ /TEST #	ERR PC	PORT #	PORT #/
10984	077346	020043	042440	051122					
10985	077354	050040	020103	050040					
10986	077362	051117	020124	020043					
10987	077370	050040	051117	020124					
10988	077376	000043							
10989	077400	020040	020040	020040	DH23:	.ASCII /	SEIZE<<CR><LF>		
10990	077406	020040	020040	020040					
10991	077414	020040	020040	042523					
10992	077422	055111	006505	012					
10993	077427	124	051505	020124		.ASCIZ /TEST #	ERR PC	PORT #	REG ADR
10994	077434	020043	042440	051122				CONTENTS/	
10995	077442	050040	020103	050040					
10996	077450	051117	020124	020043					
10997	077456	051040	043505	040440					
10998	077464	051104	041440	047117					
10999	077472	042524	052116	000123					
11000	077500	020040	020040	020040	DH26:	.ASCII /	RELSNG<<CR><LF>		
11001	077506	020040	020040	020040					
11002	077514	020040	020040	042522					
11003	077522	051514	043516	005015					
11004	077530	042524	052123	021440		.ASCIZ /TEST #	ERR PC	PORT #	
11005	077536	020040	051105	020122					
11006	077544	041520	020040	047520					
11007	077552	052122	021440	000					
11008	077557	040	020040	020040	DH31:	.ASCII /	RELSNG	RQSTNG<<CR><LF>	
11009	077564	020040	020040	020040					
11010	077572	020040	020040	051040					
11011	077600	046105	047123	020107					

11012	077606	051040	051521	047124					
11013	077614	006507	012						
11014	077617	124	051505	020124	.ASCIZ	/TEST	#	ERR PC	PORT #
11015	077624	020043	042440	051122					
11016	077632	050040	020103	050040					
11017	077640	051117	020124	020043					
11018	077646	050040	051117	020124					
11019	077654	000043							
11020	077656	042524	052123	021440	DH36:	.ASCIZ	/TEST	#	ERR PC
11021	077664	020040	051105	020122					
11022	077672	041520	020040	047520					
11023	077700	052122	021440	000					
11024	077705	124	051505	020124	DH42:	.ASCIZ	/TEST	#	ERR PC/
11025	077712	020043	042440	051122					
11026	077720	050040	000103						
11027	077734	020040	020040	020040	DH44:	.ASCII	/		RELSNG
11028	077732	020040	020040	020040					ERROR<<CR><<LF>
11029	077740	020040	020040	042522					
11030	077746	051514	043516	020040					
11031	077754	051105	047522	006522					
11032	077762	012							
11033	077763	124	051505	020124	.ASCIZ	/TEST	#	ERR PC	PORT #
11034	077770	020043	042440	051122					
11035	077776	050040	020103	050040					
11036	100004	051117	020124	020043					
11037	100012	050040	051117	020124					
11038	100020	000043							
11039	100022	020040	020040	020040	DH46:	.ASCII	/		PORT A
11040	100030	020040	020040	020040					PORT B<<CR><<LF>
11041	100036	020040	020040	047520					
11042	100044	052122	040440	020040					
11043	100052	047520	052122	041040					
11044	100060	005015							
11045	100062	042524	052123	021440	.ASCIZ	/TEST	#	ERR PC	RPDS1
11046	100070	020040	051105	020122					RMDS1/
11047	100076	041520	020040	050122					
11048	100104	051504	020061	020040					
11049	100112	046522	051504	000061					
11050	100120	042524	052123	021440	DH55:	.ASCIZ	/TEST	#	ERR PC
11051	100126	020040	051105	020122					PORT #
11052	100134	041520	020040	047520					TIMEOUT VALUE (IN MS)/
11053	100142	052122	021440	020040					
11054	100150	044524	042515	052517					
11055	100156	020124	040526	052514					
11056	100164	020105	044450	020116					
11057	100172	051515	000051						
11058	100176	051044	040515	051104	DH56:	.ASCIZ	/SRMADR		
11059	100204	000							
11060	100205	124	051505	020124	DH57:	.ASCII	/TEST	#	ERR PC
11061	100212	020043	042440	051122					PORT A
11062	100220	050040	020103	020040					PORT B/
11063	100226	020040	050040	051117					
11064	100234	020124	020101	020040					
11065	100242	020040	020040	020040					
11066	100250	050040	051117	020124					
11067	100256	102							

11089	100422	001246	001116	001242	DT7:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$GDDAT, \$BDDAT, 0
11090	100430	001240	001122	001124			
11091	100436	001126	000000				
11092	100442	001246	001116	001242	DT13:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$BDDAT, 0
11093	100450	001240	001122	001126			
11094	100456	000000					
11095	100460	001246	001116	001242	DT22:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, 0
11096	100466	001240	000000				
11097	100472	001246	001116	001242	DT23:	.WORD	TSTNUM, \$ERRPC, SEIZPT, \$BDADR, \$BDDAT, 0
11098	100500	001122	001126	000000			
11099	100506	001246	001116	001242	DT31:	.WORD	TSTNUM, \$ERRPC, SEIZPT, OPPRT, 0
11100	100514	001244	000000				
11101	100520	001246	001116	001242	DT36:	.WORD	TSTNUM, \$ERRPC, SEIZPT, 0
11102	100526	000000					
11103	100530	001246	001116	001240	DT37:	.WORD	TSTNUM, \$ERRPC, PTNBR, 0
11104	100536	000000					
11105	100540	001246	001116	000000	DT42:	.WORD	TSTNUM, \$ERRPC, 0
11106	100546	001246	001116	001170	DT46:	.WORD	TSTNUM, \$ERRPC, \$TMP2, \$TMP3, 0
11107	100554	001172	000000				
11108	100560	001246	001116	001244	DT54:	.WORD	TSTNUM, \$ERRPC, OPPRT, SEIZPT, 0
11109	100566	001242	000000				
11110	100572	001246	001116	001242	DT55:	.WORD	TSTNUM, \$ERRPC, SEIZPT, TIME, 0
11111	100600	001256	000000				
11112	100604	001304	000000		DT56:	.WORD	\$RMADR, 0
11113	100610	001246	001116	001164	DT57:	.WORD	TSTNUM, \$ERRPC, \$TMP0, \$TMP1, \$TMP2, \$TMP3, 0
11114	100616	001166	001170	001172			
11115	100624	000000					
11116							
11117	100626	000	000	000	DF1:	.BYTE	0,0,0,0,0
11118	100631	000	000				
11119	100633	000	000	000	DF5:	.BYTE	0,0,0,0,0,0
11120	100636	000	000	000			
11121	100641	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0
11122	100644	000	000	000			
11123	100647	000					
11124	100650	000	000	000	DF31:	.BYTE	0,0,0,0
11125	100653	000					
11126	100654	000	000	000	DF36:	.BYTE	0,0,0
11127	100657	000	000		DF42:	.BYTE	0,0
11128	100661	000	000	000	DF55:	.BYTE	0,0,0,1
11129	100664	001					
11130	100665	000			DF56:	.BYTE	0
11131							
11132						.EVEN	
11133							
11134							
11135							
11136							
11137							
11138							
11139							
11140							
11141							
11142							
11143	100666	003164			TSTADR:	.WORD	TST1 ; STARTING ADDRESS OF TEST 1
11144	100670	004612				.WORD	TST2 ; STARTING ADDRESS OF TEST 2

.SBTTL CONSTANTS, TABLES, ETC

;TABLE OF TEST STARTING ADDRESSES

11145	100672	006160	.WORD	TST3	: STARTING ADDRESS OF TEST 3
11146	100674	007526	.WORD	TST4	: STARTING ADDRESS OF TEST 4
11147	100676	010656	.WORD	TST5	: STARTING ADDRESS OF TEST 5
11148	100700	012006	.WORD	TST6	: STARTING ADDRESS OF TEST 6
11149	100702	012460	.WORD	TST7	: STARTING ADDRESS OF TEST 7
11150	100704	013132	.WORD	TST10	: STARTING ADDRESS OF TEST 10
11151	100706	014376	.WORD	TST11	: STARTING ADDRESS OF TEST 11
11152	100710	015642	.WORD	TST12	: STARTING ADDRESS OF TEST 12
11153	100712	016762	.WORD	TST13	: STARTING ADDRESS OF TEST 13
11154	100714	020102	.WORD	TST14	: STARTING ADDRESS OF TEST 14
11155	100716	021502	.WORD	TST15	: STARTING ADDRESS OF TEST 15
11156	100720	023102	.WORD	TST16	: STARTING ADDRESS OF TEST 16
11157	100722	024026	.WORD	TST17	: STARTING ADDRESS OF TEST 17
11158	100724	024752	.WORD	TST20	: STARTING ADDRESS OF TEST 20
11159	100726	026020	.WORD	TST21	: STARTING ADDRESS OF TEST 21
11160	100730	027066	.WORD	TST22	: STARTING ADDRESS OF TEST 22
11161	100732	031142	.WORD	TST23	: STARTING ADDRESS OF TEST 23
11162	100734	031666	.WORD	TST24	: STARTING ADDRESS OF TEST 24
11163	100736	033062	.WORD	TST25	: STARTING ADDRESS OF TEST 25
11164	100740	034256	.WORD	TST26	: STARTING ADDRESS OF TEST 26
11165	100742	035752	.WORD	TST27	: STARTING ADDRESS OF TEST 27
11166	100744	037446	.WORD	TST30	: STARTING ADDRESS OF TEST 30
11167	100746	041142	.WORD	TST31	: STARTING ADDRESS OF TEST 31
11168	100750	042636	.WORD	TST32	: STARTING ADDRESS OF TEST 32
11169	100752	044114	.WORD	TST33	: STARTING ADDRESS OF TEST 33
11170	100754	045240	.WORD	TST34	: STARTING ADDRESS OF TEST 34
11171	100756	046136	.WORD	TST35	: STARTING ADDRESS OF TEST 35
11172	100760	047034	.WORD	TST36	: STARTING ADDRESS OF TEST 36
11173	100762	050032	.WORD	TST37	: STARTING ADDRESS OF TEST 37
11174	100764	051030	.WORD	TST40	: STARTING ADDRESS OF TEST 40
11175	100766	053014	.WORD	TST41	: STARTING ADDRESS OF TEST 41
11176	100770	055000	.WORD	TST42	: STARTING ADDRESS OF TEST 42
11177	100772	056254	.WORD	TST43	: STARTING ADDRESS OF TEST 43
11178	100774	057530	.WORD	TST44	: STARTING ADDRESS OF TEST 44
11179	100776	060360	.WORD	TST45	: STARTING ADDRESS OF TEST 45
11180	101000	061210	.WORD	TST46	: STARTING ADDRESS OF TEST 46

11181					
11182					
11183			: ATTENTION BIT TABLE		
11184	101002	001	ATABIT: .BYTE	1	: ATTENTION BIT FOR DRIVE 0
11185	101003	002	.BYTE	2	: ATTENTION BIT FOR DRIVE 1
11186	101004	004	.BYTE	4	: ATTENTION BIT FOR DRIVE 2
11187	101005	010	.BYTE	10	: ATTENTION BIT FOR DRIVE 3
11188	101006	020	.BYTE	20	: ATTENTION BIT FOR DRIVE 4
11189	101007	040	.BYTE	40	: ATTENTION BIT FOR DRIVE 5
11190	101010	100	.BYTE	100	: ATTENTION BIT FOR DRIVE 6
11191	101011	200	.BYTE	200	: ATTENTION BIT FOR DRIVE 7
11192					
11193	101012	000052	MAXTN: .WORD	\$TN-1	: MAXIMUM TEST NUMBER
11194					
11195		000001	.END		

ADDRIS 071714
AORERR 071467
AOE = 001000
ASR1 001236
A*A = 100000

2094 10413#
2021 10385#
1341#
1543#
1328#
2902
3160
3628
4188
4673
5162
5487
5715
5967
6174
6379
6584
6749
6936
7327
7636
7901
8106
8454
8946
2046

2046*
2449
2906
3183
3633
4211
4679
5168
5498
5608
5969
6180
6384
6588
6753
6956
7349
7640
7903
8225
8459
9058
11184#

5043
2454
2919
3189
3656
4217
4800
5208
5500
5810
5973
6195
6407
6612
6774
6961
7354
7716
7907
8227
8482
9063

5325
2475
2923
3239
3662
4390
4805
5244
5504
5814
6036
6197
6413
6617
6779
7100
7398
7721
7983
8231
8488
9086

6999
2481
3021
3244
3802
4395
4828
5249
5528
5828
6038
6201
6429
6640
6830
7105
7402
7742
7988
8250
8505
9092

7003
2677
3026
3267
3807
4418
4834
5272
5533
5830
6042
6268
6431
6646
6834
7128
7471
7748
8009
8255
8509
9181

7008
2682
3047
3273
3830
4424
4954
5278
5638
5834
6056
6270
6435
6662
6847
7134
7473
7759
8015
8278
8615
9217

7068
2703
3053
3298
3836
4518
4959
5343
5665
5890
6058
6274
6501
6664
6851
7216
7477
7761
8026
8284
8620
9222

7069
2709
3069
3434
3975
4523
4982
5348
5669
5894
6062
6288
6503
6668
6910
7221
7499
7765
8028
8301
8754
9377

7184
2854
3073
3439
3980
4546
4988
5371
5680
5894
6118
6290
6507
6727
6912
7244
7504
7807
8032
8305
8759
9413

7185
2859
3086
3462
4003
4552
5098
5377
5682
5894
6122
6294
6521
6729
6916
7250
7548
7812
8074
8429
8912
9418

9163
2880
3090
3468
4009
4645
5134
5456
5686
5946
6146
6351
6523
6733
6930
7321
7552
7833
8079
8431
8917

9359
2886
3155
3592
4183
4650
5133
5483
5710
5952
6151
6355
6527
6747
6932
7323
7634
7839
8100
8435
8940

RTABIT 101002
RT0 = 000001
RT1 = 000002
RT2 = 000004
RT3 = 000010
RT4 = 000020
RT5 = 000040
RT6 = 000100
RT7 = 000200
R16 = 000400
R17 = 001000
BADNO 071654
BAI = 000010
BIT0 = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000

1358#
1359#
1360#
1361#
1362#
1363#
1364#
1365#
1260#
1261#
2079 10407#
1278#
1234#
1224#
1223#
1222#
1221#
1220#
1219#
1218#
1217#
1216#
1215#
1233#
1214#
1213#
1212#
1211#
1210#
1209#

1316
1234
1233
1232
1231
1230
1229
1228
1227
1226
1352 9644

9688
9651
9695
1353 9626
1354

0138	077656	1654	1696	1703	1732	1754	1775	1782	1789	1796	1803	1939	11020#		
0139	076567	1612	10917#												
0141	077705	1824	1831	11024#											
0142	077724	1838	11027#												
0146	100023	1852	11039#												
0148	076712	1619	1640	1768	1873	1921	1951	10932#							
0149	100120	1901	11050#												
0150	100176	1908	11058#												
0151	100205	1915	11060#												
0152	076766	1633	1675	1682	1689	1747	1945	10940#							
0153	001142	1507#	1991*	1999*	9665*	9687*									
0154	000174	1456#	1999												
0155	100000	1290#													
0156	000001	1351#	7523	7625	7890	7892									
0157	000010	1396#													
0158	000400	1321#	2261	2293	2384	2444	2612	2672	2849	3016	3150	3234	3346	3398	
		3429	3540	3592	3623	3797	3970	4071	4112	4155	4178	4278	4319	4362	
		4385	4490	4513	4617	4640	4749	4795	4903	4949	5075	5098	5129	5185	
		5208	5239	5338	5456	5523	5638	5705	5866	5913	6094	6141	6326	6374	
		6559	6607	7095	7211	7344	7494	7685	7711	7802	7952	7978	8069	8610	
		8749	8907	9053	9161	9212	9377	9408							
0159	004000	1381#													
0160	000200	1320#	2261	2293	2384	2444	2612	2672	2849	3016	3150	3234	3346	3398	
		3429	3540	3592	3623	3797	3970	4071	4112	4155	4178	4278	4319	4362	
		4385	4490	4513	4617	4640	4749	4795	4903	4949	5075	5098	5129	5185	
		5208	5239	5338	5456	5523	5638	5705	5866	5913	6094	6141	6326	6374	
		6559	6607	7095	7211	7311	7344	7461	7494	7685	7711	7802	7952	7978	
		8069	8610	8749	8907	9053	9181	9212	9377	9408					
0161	177570	1155#	1506	1990											
0162	010000	1344#													
0163	000001	1372#													
0164	000002	1373#													
0165	000004	1374#													
0166	000010	1375#													
0167	000020	1376#													
0168	000040	1377#													
0169	000100	1378#													
0170	000200	1379#													
0171	000400	1380#													
0172	100340	1592	1599	1648	1846	1888	11080#								
0173	100442	1627	1662	1669	1867	1881	1934	11092#							
0174	100460	1711	1741	11095#											
0175	100472	1718	1725	1811	1818	1860	1928	11097#							
0176	100354	1606	11082#												
0177	100506	1762	1839	11099#											
0178	100520	1733	1755	1776	1783	1797	1940	11101#							
0179	100530	1655	1697	1704	1790	1804	11103#								
0180	100540	1825	1832	11105#											
0181	100546	1853	11106#												
0182	100370	1620	1641	1769	1874	1922	1952	11084#							
0183	100560	1895	11103#												
0184	100572	1902	11110#												
0185	100604	1909	11112#												
0186	100610	1916	11113#												
0187	100408	11087#													
0188	100422	1613	1634	1676	1683	1690	1748	1946	11089#						

OVN	=	004000	1310#	
DVC	=	000200	1397#	
ECH	=	000100	1338#	
ECI	=	004000	1408#	
EMTVEC	=	000030	1244#	1976* 1977*
EM1		072026	1590	10433#
EM10		072433	1639	10486#
EM11		072463	1646	10494#
EM12		072547	1653	10504#
EM13		072617	1660	10512#
EM14		072704	1667	10522#
EM15		072747	1674	10529#
EM16		073026	1681	10538#
EM17		073101	1688	10547#
EM2		072047	1597	10437#
EM20		073160	1695	10556#
EM21		073240	1702	10565#
EM22		073313	1709	10574#
EM23		073400	1716	10584#
EM24		073446	1723	10592#
EM25		073525	1731	10601#
EM26		073570	1739	10608#
EM27		073655	1746	10618#
EM3		072071	1604	10442#
EM30		073733	1753	10627#
EM31		074030	1760	10639#
EM32		074105	1767	10648#
EM33		074156	1774	10656#
EM34		074260	1781	10668#
EM35		074363	1788	10681#
EM36		074442	1795	10690#
EM37		074514	1802	10698#
EM4		072153	1611	10452#
EM40		074562	1809	10706#
EM41		074637	1816	10715#
EM42		074701	1823	10722#
EM43		074767	1830	10733#
EM44		075044	1837	10742#
EM45		075121	1844	10751#
EM46		075200	1851	10760#
EM47		075247	1858	10768#
EM5		072204	1618	10458#
EM50		075335	1865	10779#
EM51		075417	1872	10789#
EM52		075506	1879	10800#
EM53		075601	1886	10811#
EM54		075662	1893	10821#
EM55		075755	1900	10832#
EM56		076022	1907	10840#
EM57		076064	1914	10847#
EM6		072252	1625	10466#
EM60		076125	1920	10854#
EM61		076171	1926	10862#
EM62		076223	1932	10868#
EM63		076271	1938	10876#
EM64		076333	1944	10883#

NED = 010000
NEM = 004000
NOATA = 000001

1287#	2161	2165	2182	2186									
1286#													
1956#	2475	2481	2703	2709	2880	2886	3047	3053	3183	3189	3267	3273	
3462#	3468	3656	3662	3830	3836	4003	4009	4211	4217	4418	4424	4546	
4552#	4673	4679	4828	4834	4982	4988	5162	5168	5272	5278	5371	5377	
5443#	5510#	5556	5561	5625#	5692#	5738	5743	5749#	5946	5952	6174	6180	
6407#	6413	6640	6646	6761#	6802	6807	6858#	6984	6989	7082#	7128	7134	
7139#	7198#	7244	7250	7255#	7377	7382	7527	7532	7696#	7742	7748	7833	
7839#	7963#	8009	8015	8100	8106	8113#	8239#	8278	8284	8289#	8443#	8482	
8488#	8493#	8643	8648	8782	8787	8901#	8940	8946	8951#	9047#	9066	9092	
9097#	9245	9250	9441	9446									

NOCL0C 071565
NOSEIZ 001252
NTRH11 071774
OFD = 000200
OM = 000001

2053	10397#												
1549#													
2098	10421#												
1406#													
1318#	2383	2447	2452	2611	2675	2680	2852	2857	3019	3024	3153	3158	
3237	3242	3345	3432	3437	3539	3626	3631	3800	3805	3973	3978	4070	
4181	4186	4277	4388	4393	4489	4516	4521	4616	4643	4648	4748	4798	
4803	4902	4952	4957	5132	5137	5242	5247	5341	5346	5526	5531	5708	
5713	5865	5916	5921	6093	6144	6149	6325	6377	6382	6558	6610	6615	
6772	6777	6954	6959	7098	7103	7214	7219	7347	7352	7497	7502	7714	
7719	7805	7810	7981	7986	8072	8077	8248	8253	8452	8457	8613	8618	
8752	8757	8910	8915	9056	9061	9215	9220	9411	9416				

OPE = 020000
OPI = 020000
OPPRT 001244

1401#													
1345#													
1546#	2370*	2598*	2817*	2984*	3332*	3526*	3719*	3892*	4057*	4264*	4476*	4603*	
4735#	4889#	5087*	5197*	5434*	5616*	5852*	6080*	6312*	6545*	7624*	7891*	8158*	
8362#	8570*	8709*	8855*	9001*	9153*	9349*	11087	11099	11108				

OR = 000200
PAR = 000010
PAT = 000020
PGE = 002000
FGM = 001000

1282#													
1335#													
1279#													
1285#													
1322#	2246	2248	2252	2278	2280	2284	2384	2444	2612	2672	2849	3016	
3150	3234	3346	3398	3429	3540	3592	3623	3797	3970	4071	4112	4155	
4178	4278	4319	4362	4385	4490	4513	4617	4640	4749	4795	4903	4949	
5075	5098	5129	5185	5208	5239	5338	5456	5523	5638	5705	5866	5913	
6094	6141	6326	6374	6559	6607	7095	7211	7344	7494	7685	7711	7802	
7952	7978	8069	8610	8749	8907	9053	9181	9212	9377	9408			
1326#	2383	2447	2452	2611	2675	2680	2852	2857	3019	3024	3153	3158	
3237	3242	3345	3432	3437	3539	3626	3631	3800	3805	3973	3978	4070	
4181	4186	4277	4388	4393	4489	4516	4521	4616	4643	4648	4748	4798	
4803	4902	4952	4957	5132	5137	5242	5247	5341	5346	5526	5531	5708	
5713	5865	5916	5921	6093	6144	6149	6325	6377	6382	6558	6610	6615	
6772	6777	6954	6959	7098	7103	7214	7219	7347	7352	7497	7502	7714	
7719	7805	7810	7981	7986	8072	8077	8248	8253	8452	8457	8613	8618	
8752	8757	8910	8915	9056	9061	9215	9220	9411	9416				

PIRQ = 177772
PIRQVE = 000240
PORTA 001224

1154#													
1248#													
1538#	2018*	2019	2023	2026	2030	2042	2045	2152	2153	2198	2199	2240	
2241	2308#	2365	2366	2380	2381	2445	2466	2468	2474	2596	2597	2598	
2625	2626	2673	2694	2696	2702	2719	2814	2815	2840	2841	2850	2871	
2873	2879	2894	2895	2984	3017	3038	3040	3046	3078	3079	3102	3103	
3132	3133	3134	3151	3172	3174	3182	3235	3256	3258	3266	3314	3330	
3331	3332	3359	3360	3401	3402	3412	3420	3421	3430	3451	3453	3461	
3508	3521	3522	3536	3537	3585	3586	3598	3599	3624	3645	3647	3655	
3704	3716	3717	3745	3746	3766	3767	3788	3789	3798	3819	3821	3829	

3877	3892	3927	3928	3971	3992	3994	4002	4052	4053	4067	4068	4106
4107	4149	4150	4169	4170	4179	4200	4202	4210	4262	4263	4264	4296
4297	4339	4340	4386	4407	4409	4417	4458	4471	4472	4486	4487	4504
4505	4514	4535	4537	4545	4585	4601	4602	4603	4641	4662	4664	4672
4717	4733	4734	4735	4758	4759	4796	4817	4819	4827	4871	4884	4885
4899	4900	4940	4941	4950	4971	4973	4981	5025	5038	5044	5045	5087
5101	5102	5112	5120	5121	5130	5151	5153	5161	5179	5180	5196	5201
5202	5214	5215	5240	5261	5263	5271	5312	5339	5360	5362	5370	5418
5434	5435	5436	5459	5460	5470	5492	5493	5514	5515	5524	5545	5547
5555	5600	5613	5614	5631	5632	5644	5645	5657	5658	5706	5727	5729
5737	5785	5795	5802	5803	5847	5848	5862	5863	5904	5905	5914	5935
5937	5945	6013	6023	6050	6051	6078	6079	6080	6142	6163	6165	6173
6189	6190	6245	6255	6262	6263	6307	6308	6322	6323	6365	6366	6375
6396	6398	6406	6478	6488	6515	6516	6543	6544	6545	6608	6629	6631
6639	6656	6657	6704	6714	6721	6722	6770	6791	6793	6801	6822	6823
6889	6899	6904	6905	6952	6973	6975	6983	7014	7015	7052	7060	7096
7117	7119	7127	7168	7179	7188	7189	7190	7202	7203	7212	7233	7235
7243	7291	7299	7300	7301	7335	7336	7345	7366	7368	7376	7441	7495
7516	7518	7526	7536	7537	7621	7622	7702	7703	7712	7733	7735	7741
7803	7824	7826	7832	7891	7979	8000	8002	8008	8020	8021	8060	8061
8070	8091	8093	8099	8143	8155	8156	8182	8183	8246	8267	8269	8277
8313	8347	8362	8374	8375	8450	8471	8473	8481	8497	8498	8554	8567
8568	8611	8632	8634	8642	8693	8709	8733	8734	8750	8771	8773	8781
8839	8855	8856	8857	8908	8929	8931	8939	8985	8998	8999	9009	9010
9054	9075	9077	9085	9139	9150	9151	9159	9160	9168	9169	9174	9175
9187	9188	9213	9234	9236	9244	9335	9349	9352	9353	9360	9361	9380
9381	9391	9399	9400	9409	9430	9432	9440					
2029	10389#											
1539#	2023*	2024*	2025*	2028*	2036	2173	2174	2217	2218	2272	2273	2310
2368	2369	2370	2397	2398	2450	2462	2463	2480	2491	2593	2594	2608
2609	2678	2690	2691	2708	2817	2855	2867	2868	2885	2911	2912	2935
2936	2981	2982	3007	3008	3022	3034	3035	3052	3061	3062	3156	3168
3169	3188	3216	3217	3218	3240	3252	3253	3272	3318	3327	3328	3342
3343	3391	3392	3404	3405	3435	3447	3448	3467	3512	3524	3525	3526
3553	3554	3595	3596	3606	3614	3615	3629	3641	3642	3661	3708	3719
3754	3755	3803	3815	3816	3835	3881	3889	3890	3918	3919	3939	3940
3961	3962	3976	3988	3989	4008	4055	4056	4057	4089	4090	4132	4133
4184	4196	4197	4216	4259	4260	4274	4275	4313	4314	4356	4357	4376
4377	4391	4403	4404	4423	4462	4474	4475	4476	4519	4531	4532	4551
4589	4598	4599	4613	4614	4631	4632	4646	4658	4659	4678	4721	4730
4731	4745	4746	4786	4787	4801	4813	4814	4833	4875	4887	4888	4889
4912	4913	4955	4967	4968	4987	5029	5052	5053	5059	5060	5086	5091
5092	5104	5105	5135	5147	5148	5167	5197	5211	5212	5222	5230	5231
5245	5257	5258	5277	5316	5344	5356	5357	5376	5422	5431	5432	5449
5450	5462	5463	5475	5476	5529	5541	5542	5560	5604	5616	5617	5618
5641	5642	5652	5674	5675	5696	5697	5711	5723	5724	5742	5790	5822
5823	5850	5851	5852	5919	5931	5932	5951	5961	5962	6018	6030	6031
6075	6076	6090	6091	6132	6133	6147	6159	6160	6179	6250	6282	6283
6310	6311	6312	6380	6392	6393	6412	6423	6424	6483	6495	6496	6540
6541	6555	6556	6598	6599	6613	6625	6626	6645	6709	6741	6742	6775
6787	6788	6806	6839	6840	6894	6924	6925	6957	6969	6970	6988	7018
7019	7056	7063	7072	7073	7074	7086	7087	7101	7113	7114	7133	7172
7176	7217	7229	7230	7249	7295	7350	7362	7363	7381	7386	7387	7445
7449	7450	7451	7485	7486	7500	7512	7513	7531	7624	7717	7729	7730
7747	7753	7754	7793	7794	7809	7820	7821	7838	7888	7889	7969	7970
7984	7996	7997	8014	8075	8087	8088	8105	8147	8156	8170	8171	8251

PORTA1 071511
PORTB 001226

GO1

CZRMGB0 RMO3/2 DU POR LGC 1
 CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 214
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0213

		8263	8264	8283	8293	8294	8351	8359	8360	8386	8387	8455	8467	8468
		8487	8517	8558	8570	8594	8595	8616	8628	8629	8647	8697	8706	8707
		8755	8767	8768	8786	8843	8852	8853	8863	8864	8913	8925	8926	8945
		8989	9001	9002	9003	9059	9071	9072	9091	9143	9153	9156	9157	9164
		9165	9184	9185	9195	9203	9204	9218	9230	9231	9249	9339	9346	9347
		9355	9356	9364	9365	9370	9371	9383	9384	9414	9426	9427	9445	
PORTBI	071537	2035	10393*											
PORTC	001230	1540*	2042*	2043*	2044*	5037	5320							
PRO	= 000000	1171*												
PR1	= 000040	1172*												
PR2	= 000100	1173*												
PR3	= 000140	1174*												
PR4	= 000200	1175*												
PR5	= 000240	1176*												
PR6	= 000300	1177*												
PR7	= 000340	1178*												
PS	= 177776	1151*	1152	2060*	2350*	2420*	2437*	2578*	2648*	2665*				
PSEL	= 002000	1262*												
PSW	= 177776	1152*												
PTNBR	001240	1544*												
		2480*	2153*	2174*	2199*	2218*	2241*	2273*	2369*	2381*	2398*	2462*	2466*	2474*
		2879*	2491*	2597*	2609*	2626*	2690*	2694*	2702*	2708*	2719*	2841*	2867*	2871*
		3133*	2885*	2895*	2912*	2936*	3008*	3034*	3038*	3046*	3052*	3062*	3079*	3103*
		3392*	3168*	3172*	3182*	3188*	3217*	3252*	3256*	3266*	3272*	3331*	3343*	3360*
		3596*	3402*	3405*	3412*	3421*	3447*	3451*	3461*	3467*	3525*	3537*	3554*	3586*
		3819*	3599*	3606*	3615*	3641*	3645*	3655*	3661*	3746*	3755*	3767*	3789*	3815*
		4090*	3829*	3835*	3919*	3928*	3940*	3962*	3988*	3992*	4002*	4008*	4056*	4068*
		4340*	4107*	4133*	4150*	4170*	4196*	4200*	4210*	4216*	4263*	4275*	4297*	4314*
		4551*	4357*	4377*	4403*	4407*	4417*	4423*	4475*	4487*	4505*	4531*	4535*	4545*
		4817*	4602*	4614*	4632*	4658*	4662*	4672*	4678*	4734*	4746*	4759*	4787*	4813*
		5060*	4827*	4833*	4888*	4900*	4913*	4941*	4967*	4971*	4981*	4987*	5045*	5053*
		5215*	5092*	5102*	5105*	5112*	5121*	5147*	5151*	5161*	5167*	5180*	5202*	5212*
		5460*	5222*	5231*	5257*	5261*	5271*	5277*	5356*	5360*	5370*	5376*	5436*	5450*
		5645*	5463*	5470*	5476*	5493*	5515*	5541*	5545*	5555*	5560*	5618*	5632*	5642*
		5905*	5652*	5658*	5675*	5697*	5723*	5727*	5737*	5742*	5803*	5823*	5851*	5863*
		6173*	5931*	5935*	5945*	5951*	5962*	6031*	6051*	6079*	6091*	6133*	6159*	6163*
		6496*	6179*	6190*	6263*	6283*	6311*	6323*	6366*	6392*	6396*	6406*	6412*	6424*
		6791*	6516*	6544*	6556*	6599*	6625*	6629*	6639*	6645*	6657*	6722*	6742*	6787*
		7073*	6801*	6806*	6823*	6840*	6905*	6925*	6969*	6973*	6983*	6988*	7015*	7019*
		7336*	7087*	7113*	7117*	7127*	7133*	7189*	7203*	7229*	7233*	7243*	7249*	7300*
		7703*	7362*	7366*	7376*	7381*	7387*	7450*	7486*	7512*	7516*	7526*	7531*	7537*
		8000*	7729*	7733*	7741*	7747*	7754*	7794*	7820*	7824*	7832*	7838*	7970*	7996*
		8277*	8008*	8014*	8021*	8061*	8087*	8091*	8099*	8105*	8171*	8183*	8263*	8267*
		8642*	8283*	8294*	8375*	8387*	8467*	8471*	8481*	8487*	8498*	8595*	8628*	8632*
		9003*	8647*	8734*	8767*	8771*	8781*	8786*	8857*	8864*	8925*	8929*	8939*	8945*
		9195*	9010*	9071*	9075*	9085*	9091*	9157*	9160*	9165*	9169*	9175*	9185*	9188*
		9391*	9204*	9230*	9234*	9244*	9249*	9353*	9356*	9361*	9365*	9371*	9381*	9384*
		1243*	9400*	9426*	9430*	9440*	9445*	11080	11084	11089	11092	11095	11103	
PWRVEC	= 000024	10165												
RDCHR	= 104410	10238	10357*											
RDLIN	= 104411	2017	10358*											
RDOCT	= 104412	1259*	2073	2099	10359*									
RDY	= 000200	1550*												
RELERP	001254	3231*	2441*	2471*	2669*	2699*	2846*	2876*	2891	3013*	3043*	3058	3147*	3177*
		4205*	3261*	3397*	3426*	3456*	3591*	3620*	3650*	3794*	3824*	3967*	3997*	4175*
		5156*	4382*	4412*	4510*	4540*	4637*	4667*	4792*	4822*	4946*	4976*	5097*	5126*
			5207*	5236*	5266*	5335*	5365*	5455*	5520*	5550*	5637*	5702*	5732*	5910*

HO1

CZRMGB0 RM03-2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 215
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0214

RELOK = 000001

RESREG= 104414
RESVEC= 000010
RMAS = 000016
RMB4 = 000004
RMCS1 = 000000

RMCS2 = 000010

5940*	6138*	6168*	6371*	6401*	6604*	6634*	6766*	6796*	6811	6948*	6978*	6993
7092*	7122*	7208*	7238*	7341*	7371*	7491*	7521*	7708*	7738*	7799*	7829*	7975*
8005*	8066*	8096*	8242*	8272*	8446*	8476*	8607*	8637*	8746*	8776*	8904*	8954*
9050*	9080*	9180*	9209*	9239*	9376*	9405*	9435*					
1956*	2472	2700	2877	3044	3111*	3178	3262	3457	3651	3825	3998	4206
4413	4541	4668	4823	4977	5157	5267	5366	5551	5733	5941	6169	6402
6635	6797	6979	7123	7239	7372	7522	7739	7830	8006	8097	8273	8477
8638	8777	8935	9081	9240	9436							
10361*												
1238*												
1436*	5043*	5327*	6999*	7002	7006	7066	7076	7182	7192	9163*	9359*	
1431*												
1429*	2818*	2842*	2985*	3009*	3139*	3178*	3179*	3223*	3262*	3263*	3316*	3317*
3320*	3321*	3361*	3375	3376	3381*	3393*	3422*	3457*	3458*	3510*	3511*	3514*
3515*	3555*	3569	3570	3575*	3587*	3616*	3651*	3652*	3706*	3707*	3710*	3711*
3725*	3727	3728	3747*	3782*	3790*	3825*	3826*	3879*	3880*	3883*	3884*	3898*
3900	3901	3920*	3955*	3963*	3998*	3999*	4084*	4171*	4206*	4207*	4291*	4378*
4413*	4414*	4460*	4461*	4464*	4465*	4473	4506*	4541*	4542*	4587*	4588*	4591*
4592*	4600	4633*	4668*	4669*	4719*	4720*	4723*	4724*	4732	4765	4766	4788*
4823*	4824*	4873*	4874*	4877*	4878*	4886	4919	4920	4942*	4977*	4978*	5027*
5028*	5031*	5032*	5093*	5122*	5157*	5158*	5203*	5232*	5267*	5268*	5314*	5315*
5318*	5319*	5366*	5367*	5420*	5421*	5424*	5425*	5451*	5516*	5551*	5552*	5602*
5603*	5606*	5607*	5633*	5698*	5733*	5734*	5879*	5906*	5941*	5942*	6107*	6134*
6169*	6170*	6340*	6367*	6402*	6403*	6573*	6600*	6635*	6636*	6797*	6798*	6799*
6980*	7054*	7055*	7058*	7059*	7088*	7123*	7124*	7170*	7171*	7174*	7175*	7204*
7239*	7240*	7293*	7294*	7297*	7298*	7306*	7337*	7372*	7373*	7443*	7444*	7447*
7448*	7456*	7487*	7522*	7523*	7662*	7663*	7704*	7788*	7789*	7795*	7929*	7930*
7971*	8055*	8056*	8062*	8145*	8146*	8149*	8150*	8273*	8274*	8315*	8316*	8349*
8350*	8353*	8354*	8477*	8478*	8519*	8520*	8556*	8557*	8560*	8561*	8638*	8639*
8695*	8696*	8699*	8700*	8777*	8778*	8841*	8842*	8845*	8846*	8864*	8935*	8936*
8987*	8988*	8991*	8992*	9015*	9081*	9082*	9141*	9142*	9145*	9146*	9176*	9205*
9240*	9241*	9269	9270	9275*	9286	9287	9292*	9337*	9338*	9341*	9342*	9372*
9401*	9436*	9437*	9465	9466	9471*	9482	9483	9488*	9489*			
1433*	2147*	2152*	2156	2157	2172*	2173*	2177	2178	2193*	2198*	2217*	2240*
2272*	2308*	2310*	2365*	2368*	2380*	2382*	2445*	2450*	2463*	2468*	2593*	2596*
2608*	2625*	2673*	2678*	2691*	2696*	2814*	2834*	2840*	2850*	2855*	2868*	2873*
2894*	2911*	2935*	2981*	3001*	3007*	3017*	3022*	3035*	3040*	3061*	3078*	3102*
3132*	3151*	3156*	3169*	3174*	3216*	3235*	3240*	3253*	3258*	3314*	3318*	3327*
3330*	3342*	3359*	3391*	3401*	3404*	3420*	3430*	3435*	3448*	3453*	3508*	3512*
3521*	3524*	3536*	3553*	3585*	3595*	3598*	3614*	3624*	3629*	3642*	3647*	3704*
3708*	3716*	3744*	3745*	3754*	3766*	3788*	3798*	3803*	3816*	3821*	3877*	3881*
3889*	3917*	3918*	3927*	3939*	3961*	3971*	3976*	3989*	3994*	4052*	4055*	4067*
4089*	4106*	4127*	4132*	4149*	4169*	4179*	4184*	4197*	4202*	4259*	4262*	4274*
4296*	4313*	4334*	4339*	4356*	4376*	4386*	4391*	4404*	4409*	4458*	4462*	4471*
4474*	4486*	4504*	4514*	4519*	4532*	4537*	4585*	4589*	4598*	4601*	4613*	4631*
4641*	4646*	4659*	4664*	4717*	4721*	4730*	4733*	4745*	4758*	4786*	4796*	4801*
4814*	4819*	4871*	4875*	4884*	4887*	4899*	4912*	4940*	4950*	4955*	4968*	4973*
5025*	5029*	5037*	5044*	5052*	5059*	5091*	5101*	5104*	5120*	5130*	5135*	5148*
5153*	5179*	5201*	5211*	5214*	5230*	5240*	5245*	5258*	5263*	5312*	5316*	5320*
5339*	5344*	5357*	5362*	5418*	5422*	5431*	5435*	5449*	5459*	5462*	5475*	5492*
5514*	5524*	5529*	5542*	5547*	5600*	5604*	5613*	5617*	5631*	5641*	5644*	5657*
5674*	5696*	5706*	5711*	5724*	5729*	5785*	5790*	5795*	5802*	5822*	5847*	5850*
5862*	5904*	5914*	5919*	5932*	5937*	5961*	6013*	6018*	6023*	6030*	6050*	6075*
6078*	6090*	6132*	6142*	6147*	6160*	6165*	6189*	6245*	6250*	6255*	6262*	6282*
6307*	6310*	6322*	6365*	6375*	6380*	6393*	6398*	6423*	6478*	6483*	6488*	6495*
6515*	6540*	6543*	6555*	6598*	6608*	6613*	6626*	6631*	6656*	6704*	6709*	6714*

6721*	6741*	6770*	6775*	6788*	6793*	6817*	6822*	6839*	6889*	6894*	6899*	6904*
6924*	6952*	6957*	6970*	6975*	7014*	7018*	7052*	7056*	7060*	7063*	7072*	7086*
7096*	7101*	7114*	7119*	7168*	7172*	7176*	7179*	7188*	7202*	7212*	7217*	7230*
7235*	7291*	7295*	7299*	7335*	7345*	7350*	7363*	7368*	7386*	7441*	7445*	7449*
7485*	7495*	7500*	7513*	7518*	7536*	7621*	7702*	7712*	7717*	7730*	7735*	7753*
7793*	7803*	7808*	7821*	7826*	7888*	7969*	7979*	7984*	7997*	8002*	8020*	8060*
8070*	8075*	8088*	8093*	8143*	8147*	8155*	8170*	8182*	8246*	8251*	8264*	8269*
8293*	8313*	8347*	8351*	8359*	8374*	8386*	8450*	8455*	8468*	8473*	8497*	8517*
8554*	8558*	8567*	8594*	8611*	8616*	8629*	8634*	8693*	8697*	8706*	8733*	8750*
8755*	8768*	8773*	8839*	8843*	8852*	8856*	8863*	8908*	8913*	8926*	8931*	8985*
8989*	8998*	9002*	9009*	9054*	9059*	9072*	9077*	9139*	9143*	9150*	9156*	9159*
9164*	9168*	9174*	9184*	9187*	9203*	9213*	9218*	9231*	9236*	9335*	9339*	9346*
9352*	9355*	9360*	9364*	9370*	9380*	9383*	9399*	9409*	9414*	9427*	9432*	
1432*	2367*	2407	2503	2595*	2635	2731						
1438*												
1443*	2402	2528	2630	2756								
1434*	2154	2175	2243	2244	2258	2259	2275	2276	2290	2291	2371	2373
2382	2414	2442	2446	2451	2599	2601	2610	2642	2670	2674	2679	2816*
2820	2821	2847	2851	2856	2897	2898	2914	2915	2937	2983*	2987	2988
3014	3018	3023	3064	3065	3081	3082	3104	3148	3152	3157	3232	3236
3241	3315*	3319*	3329*	3333	3335	3344	3367	3368	3399	3403	3406	3427
3431	3436	3509*	3513*	3523*	3527	3529	3538	3561	3562	3593	3597	3600
3621	3625	3630	3705*	3709*	3718*	3757	3758	3795	3799	3804	3878*	3882*
3891*	3930	3931	3968	3972	3977	4054*	4058	4060	4069	4092	4093	4109
4110	4135	4136	4152	4153	4176	4180	4185	4261*	4265	4267	4276	4299
4300	4316	4317	4342	4343	4359	4360	4383	4387	4392	4459*	4463*	4477
4479	4488	4511	4515	4520	4586*	4590*	4604	4606	4615	4638	4642	4647
4718*	4722*	4736	4738	4747	4793	4797	4802	4872*	4876*	4890	4892	4901
4947	4951	4956	5026*	5030*	5050	5054	5061	5072	5073	5099	5103	5106
5127	5131	5136	5182	5183	5209	5213	5216	5237	5241	5246	5313*	5317*
5336	5340	5345	5419*	5423*	5433*	5441*	5457	5461	5464	5479	5479	5495
5496	5521	5525	5530	5601*	5605*	5615*	5623*	5639	5643	5646	5660	5661
5677	5678	5703	5707	5712	5786	5791	5796	5805	5806	5825	5826	5849*
5853	5855	5864	5885	5886	5911	5915	5920	5964	5965	6014	6019	6024
6033	6034	6053	6054	6077*	6081	6083	6092	6113	6114	6139	6143	6148
6192	6193	6246	6251	6256	6265	6266	6285	6286	6309*	6313	6315	6324
6346	6347	6372	6376	6381	6426	6427	6479	6484	6489	6498	6499	6518
6519	6542*	6546	6548	6557	6579	6580	6605	6609	6614	6659	6660	6705
6710	6715*	6724	6725	6744	6745	6767	6771	6776	6825	6826	6842	6843
6890	6895	6900	6907	6908	6927	6928	6949	6953	6958	7016	7020	7053*
7057*	7064	7075*	7093	7097	7102	7169*	7173*	7180	7191*	7209	7213	7218
7292*	7296*	7311	7318	7319	7342	7346	7351	7393	7394	7442*	7446*	7461
7468	7469	7492	7496	7501	7543	7544	7631	7632	7646	7647	7667	7668
7682	7683	7709	7713	7718	7756	7757	7771	7772	7800	7804	7809	7898
7899	7913	7914	7934	7935	7949	7950	7976	7980	7985	8023	8024	8038
8039	8067	8071	8076	8144*	8148*	8157*	8176	8189	8190	8222	8223	8243
8247	8252	8296	8297	8314*	8348*	8352*	8361*	8380	8393	8394	8426	8427
8447	8451	8456	8500	8501	8518*	8555*	8559*	8569*	8593	8596	8608	8612
8617	8694*	8698*	8708*	8732	8735	8747	8751	8756	8840*	8844*	8854*	8862*
8880	8881	8905	8909	8914	8986*	8990*	9000*	9008*	9026	9027	9051	9055
9060	9140*	9144*	9152*	9182	9186	9189	9210	9214	9219	9336*	9340*	9348*
9378	9382	9385	9406	9410	9415							
1440*	2201	2202	2220	2221	2406	2508	2634	2736				
1446*	2400	2538	2628	2766								
1447*	2399	2543	2627	2771								
1435*	2409	2493	2637	2721	3724*	3769	3770	3897*	3942	3943	5788*	5789*

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

RMDT = 000026
RMEC1 = 000044
RMEC2 = 000046
RMEF1 = 000014

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 221
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0220

	8580*	8581*	8588*	8598	8719*	8720	8727*	8737	8874*	8895	9020*	9041	9585
WCE = 040000	9587*	9589*											
WCF = 000040	1289*												
WLE = 004000	1337*												
WRL = 004000	1343*												
	1324*	2447	2452	2675	2680	2852	2857	3019	3024	3153	3158	3237	3242
	3432	3437	3626	3631	3800	3805	3973	3978	4181	4186	4388	4393	4516
	4521	4643	4648	4798	4803	4952	4957	5132	5137	5242	5247	5341	5346
	5526	5531	5708	5713	5916	5921	6144	6149	6377	6382	6610	6615	6772
	6777	6754	6959	7098	7103	7214	7219	7347	7352	7497	7502	7714	7719
	7805	7810	7981	7986	8072	8077	8248	8253	8452	8457	8613	8618	8752
	8757	8910	8915	9056	9061	9215	9220	9411	9416				
\$AUTOB 001134	1503*	2013*	10070	10218									
\$BDADR 001122	1498*	2157*	2158*	2178*	2179*	2202*	2203*	2221*	2222*	2244*	2245*	2259*	2260*
	2276*	2277*	2291*	2292*	2372*	2373*	2442*	2443*	2492*	2493*	2497*	2498*	2502*
	2503*	2507*	2508*	2512*	2513*	2517*	2518*	2522*	2523*	2527*	2528*	2532*	2533*
	2537*	2538*	2542*	2543*	2600*	2601*	2670*	2671*	2720*	2721*	2725*	2726*	2730*
	2731*	2735*	2736*	2740*	2741*	2745*	2746*	2750*	2751*	2755*	2756*	2760*	2761*
	2765*	2766*	2770*	2771*	2821*	2822*	2847*	2848*	2898*	2899*	2915*	2916*	2988*
	2589*	3014*	3015*	3065*	3066*	3082*	3083*	3148*	3149*	3232*	3233*	3334*	3335*
	3368*	3369*	3376*	3377*	3399*	3400*	3427*	3428*	3528*	3529*	3562*	3563*	3570*
	3571*	3593*	3594*	3621*	3622*	3728*	3729*	3758*	3759*	3770*	3771*	3795*	3796*
	3901*	3902*	3931*	3932*	3943*	3944*	3968*	3969*	4059*	4060*	4093*	4094*	4110*
	4111*	4136*	4137*	4153*	4154*	4176*	4177*	4266*	4267*	4300*	4301*	4317*	4318*
	4343*	4344*	4360*	4361*	4383*	4384*	4478*	4479*	4511*	4512*	4605*	4606*	4638*
	4639*	4737*	4738*	4766*	4767*	4793*	4794*	4891*	4892*	4920*	4921*	4947*	4948*
	5073*	5074*	5099*	5100*	5127*	5128*	5183*	5184*	5209*	5210*	5237*	5238*	5336*
	5337*	5457*	5458*	5479*	5480*	5496*	5497*	5521*	5522*	5639*	5640*	5661*	5662*
	5678*	5679*	5703*	5704*	5806*	5807*	5826*	5827*	5854*	5855*	5886*	5887*	5911*
	5912*	5965*	5966*	6034*	6035*	6054*	6055*	6082*	6083*	6114*	6115*	6139*	6140*
	6193*	6194*	6266*	6267*	6286*	6287*	6314*	6315*	6347*	6348*	6372*	6373*	6427*
	6428*	6499*	6500*	6519*	6520*	6547*	6548*	6580*	6591*	6605*	6606*	6660*	6661*
	6725*	6726*	6745*	6746*	6767*	6768*	6826*	6827*	6843*	6844*	6908*	6909*	6928*
	6929*	6949*	6950*	7005*	7006*	7066*	7067*	7093*	7094*	7182*	7183*	7209*	7210*
	7319*	7320*	7342*	7343*	7394*	7395*	7469*	7470*	7492*	7493*	7544*	7545*	7632*
	7633*	7647*	7648*	7668*	7669*	7683*	7684*	7709*	7710*	7757*	7758*	7772*	7773*
	7800*	7801*	7899*	7900*	7914*	7915*	7935*	7936*	7950*	7951*	7976*	7977*	8024*
	8025*	8039*	8040*	8067*	8068*	8190*	8191*	8209*	8210*	8223*	8224*	8243*	8244*
	8297*	8298*	8394*	8395*	8413*	8414*	8427*	8428*	8447*	8448*	8501*	8502*	8608*
	8609*	8747*	8748*	8881*	8882*	8905*	8906*	9027*	9028*	9051*	9052*	9182*	9183*
	9210*	9211*	9262*	9263*	9270*	9271*	9279*	9280*	9287*	9288*	9378*	9379*	9406*
	9407*	9458*	9459*	9466*	9467*	9475*	9476*	9483*	9484*	11080	11082	11084	11087
\$BDADR 001126	11089	11092	11097										
	1500*	2156*	2160	2164	2177*	2181	2185	2201*	2205	2207	2211	2220*	2224
	2226	2230	2243*	2247	2251	2258*	2262	2266	2275*	2279	2283	2290*	2294
	2298	2311*	2312	2371*	2375	2382*	2383*	2387	2461*	2467*	2473*	2475*	2476
	2479*	2481*	2482	2494*	2499*	2504*	2509*	2514*	2519*	2524*	2529*	2534*	2539*
	2544*	2599*	2603	2610*	2611*	2615	2689*	2695*	2701*	2703*	2704	2707*	2709*
	2710	2722*	2727*	2732*	2737*	2742*	2747*	2752*	2757*	2762*	2767*	2772*	2820*
	2824	2828	2866*	2872*	2878*	2880*	2881	2884*	2886*	2887	2897*	2901	2905
	2914*	2918	2922	2987*	2991	2995	3033*	3039*	3045*	3047*	3048	3051*	3053*
	3054	3064*	3068	3072	3081*	3085	3089	3167*	3173*	3181*	3183*	3184	3187*
	3189*	3190	3251*	3257*	3265*	3267*	3268	3271*	3273*	3274	3333*	3337	3344*
	3345*	3349	3367*	3371	3375*	3378	3406*	3411*	3413	3446*	3452*	3460*	3462*
	3463	3466*	3468*	3469	3527*	3531	3538*	3539*	3543	3561*	3565	3569*	3572
	3600*	3605*	3607	3640*	3646*	3654*	3656*	3657	3660*	3662*	3663	3727*	3731

3735	3757*	3761	3769*	3773	3814*	3820*	3828*	3830*	3831	3834*	3836*	3837
3900*	3904	3908	3930*	3934	3942*	3946	3987*	3993*	4001*	4003*	4004	4007*
4009*	4010	4058*	4062	4069*	4070*	4074	4092*	4096	4100	4109*	4113	4117
4135*	4139	4143	4152*	4156	4160	4195*	4201*	4209*	4211*	4212	4215*	4217*
4218	4265*	4269	4276*	4277*	4281	4299*	4303	4307	4316*	4320	4324	4342*
4346	4350	4359*	4363	4367	4402*	4408*	4416*	4418*	4419	4422*	4424*	4425*
4477*	4481	4488*	4489*	4493	4530*	4536*	4544*	4546*	4547	4550*	4552*	4553
4604*	4608	4615*	4616*	4620	4657*	4663*	4671*	4673*	4674	4677*	4679*	4680
4736*	4740	4747*	4748*	4752	4765*	4769	4773	4812*	4818*	4826*	4828*	4829
4832*	4834*	4835	4890*	4894	4901*	4902*	4906	4919*	4923	4927	4966*	4972*
4980*	4982*	4983	4986*	4988*	4989	5072*	5076	5080	5106*	5111*	5113	5146*
5152*	5160*	5162*	5163	5166*	5168*	5169	5182*	5186	5190	5216*	5221*	5223
5256*	5262*	5270*	5272*	5273	5276*	5278*	5279	5355*	5361*	5369*	5371*	5372
5375*	5377*	5378	5464*	5469*	5471	5478*	5482	5486	5495*	5499	5503	5540*
5546*	5554*	5556	5559*	5561	5646*	5651*	5653	5660*	5664	5668	5677*	5681
5685	5722*	5728*	5736*	5738	5741*	5743	5805*	5809	5813	5825*	5829	5833
5853*	5857	5864*	5865*	5869	5885*	5889	5893	5930*	5936*	5944*	5946*	5947
5950*	5952*	5953	5964*	5968	5972	6033*	6037	6041	6053*	6057	6061	6081*
6085	6092*	6093*	6097	6113*	6117	6121	6158*	6164*	6172*	6174*	6175	6178*
6180*	6181	6192*	6196	6200	6265*	6269	6273	6295*	6289	6293	6313*	6317
6324*	6325*	6329	6346*	6350	6354	6391*	6397*	6405*	6407*	6408	6411*	6413*
6414	6426*	6430	6434	6498*	6502	6506	6518*	6522	6526	6546*	6550	6557*
6558*	6562	6579*	6583	6587	6624*	6630*	6638*	6640*	6641	6644*	6646*	6647
6659*	6663	6667	6724*	6728	6732	6744*	6748	6752	6786*	6792*	6800*	6802
6905*	6807	6825*	6829	6833	6842*	6846	6850	6907*	6911	6915	6927*	6931
6935	6968*	6974*	6982*	6984	6987*	6989	7002*	7003	7007	7076*	7077	7112*
7118*	7126*	7128*	7129	7132*	7134*	7135	7192*	7193	7228*	7234*	7242*	7244*
7245	7248*	7250*	7251	7318*	7322	7326	7361*	7367*	7375*	7377	7380*	7382
7393*	7397	7401	7468*	7472	7476	7511*	7517*	7525*	7527	7530*	7532	7543*
7547	7551	7631*	7635	7639	7646*	7650	7654	7667*	7671	7675	7682*	7686
7690	7728*	7734*	7740*	7742*	7743	7746*	7748*	7749	7756*	7760	7764	7771*
7775	7779	7819*	7825*	7831*	7833*	7834	7837*	7839*	7840	7898*	7902	7906
7913*	7917	7921	7934*	7938	7942	7949*	7953	7957	7995*	8001*	8007*	8009*
8010	8013*	8015*	8016	8023*	8027	8031	8038*	9042	8046	8086*	8092*	8098*
8100*	8101	8104*	8106*	8107	8189*	8193	8197	8208*	8212	8222*	8226	8230
8262*	8269*	8276*	8278*	8279	8282*	8284*	8285	8296*	8300	8304	8393*	8397
8401	8412*	8416	8426*	8430	8434	8466*	8472*	8480*	8482*	8483	8486*	8488*
8489	8500*	8504	8508	8627*	8633*	8641*	8643	8646*	8648	8766*	8772*	8780*
8782	8785*	8787	8880*	8884	8924*	8930*	8938*	8940*	8941	8944*	8946*	8947
9026*	9030	9070*	9076*	9084*	9086*	9087	9090*	9092*	9093	9189*	9194*	9196
9229*	9235*	9243*	9245	9248*	9250	9261*	9265	9269*	9272	9278*	9282	9286*
9289	9385*	9390*	9392	9425*	9431*	9439*	9441	9444*	9446	9457*	9461	9465*
9468	9474*	9478	9482*	9485	11080	11082	11084	11087	11089	11092	11097	
1526*	9690	9710	10023	10212								
9802*	9812*	9819	9828*	9833*								
10056*	10356											
1486*	1967	1968	1976	1980	1981	1982						
1518*	1519*											
1518*	1519*											
1516*	1518											
1519*	1520*	1521*	1522*	1523*	1524*							
10077	10213*											
10094	10187	10212*										
1528*	2041	2061	2097	9533	9698	9710	9718	9737	9742	9762	9801	9836
10105	10192	10212	10272									
9937	9971	9979*										

SBELL 001202
 SCHARC 067176
 SCKSWP 070126
 SCMTAG 001100
 SCM1 = 000001
 SCM2 = 000002
 SCM3 = 000001
 SCM4 = 000005
 SCNTLG 070770
 SCNTLU 070763
 SCPLF 001207
 SDB_V 067644

SSVLAD 066410
SSVPC = 000200
SSWR = 166000

9635	9660*												
1463*	1468												
1115*	1126	1131	1132	1133	1134	1135	1136	1137	1524	1525	1526	1981	
1982	1984	1985	2145	2348	2576	2800	2967	3130	3214	3309	3503	3699	
3872	4045	4252	4453	4580	4712	4866	5020	5307	5413	5595	5779	6007	
6238	6471	6698	6883	7047	7163	7286	7436	7611	7878	8138	8342	8549	
8688	8834	8980	9134	9330	9494	9502	9510	9537	9543	9545	9617	9618	
9619	9620	9621	9626	9638	9640	9641	9642	9649	9650	9651	9662	9665	
9668	9669*	9675	9676	9677	9678	9679	9688	9695	9700	9704	9710		

SSWRMK = 000000
\$TIMES 00:1176

9621	1524*	1981*	2145*	2348*	2576*	2800*	2967*	3130*	3214*	3309*	3503*	3699*	3872*
1524*	4045*	4252*	4453*	4580*	4712*	4866*	5020*	5307*	5413*	5595*	5779*	6007*	6238*
6471*	6698*	6883*	7047*	7163*	7286*	7436*	7611*	7878*	8138*	8342*	8549*	8688*	
8834*	8980*	9134*	9330*	9510*	9649*	9656	9659*	9668					
1509*	9983	10004	10013	10032	10060	10087							
9984*	9999*	10021	10038*	10144	10146*								
2005	2366	9999*	10073										
9988*	10046	10149											
9985*	10000*	10001	10044*	10045*	10046	10048*							
9986*	10001*	10147	10148*	10149	10151*								
9987*	10000	10048	10151										
1508*	9983	10005*	10028*	10030	10036*	10058	10074*	10084	10108*				
10002	10013*												

\$TKB 001146
\$TKCNT 067654
\$TKINT 067664
\$TKQEN= 067663
\$TKQIN 067656
\$TKQOU 067660
\$TKQSR 067662
\$TKS 001144
\$TKSRV 067734
\$MPO 001164

1519*	2160*	2161*	2162	2181*	2182*	2183	2207*	2208*	2209	2226*	2227*	2228	
2247*	2248*	2249	2262*	2263*	2264	2279*	2280*	2281	2294*	2295*	2296	2387*	
2388*	2389	2448*	2449*	2455	2457	2464	2615*	2616*	2617	2676*	2677*	2683	
2685	2692	2824*	2825*	2826	2853*	2854*	2860	2862	2869	2901*	2902*	2903	
2918*	2919*	2920	2991*	2992*	2993	3020*	3021*	3027	3029	3036	3068*	3069*	
3070	3085*	3086*	3087	3154*	3155*	3161	3163	3170	3238*	3239*	3245	3247	
3254	3349*	3350*	3351	3403*	3408	3411	3433*	3434*	3440	3442	3449	3543*	
3544*	3545	3597*	3602	3605	3627*	3628*	3634	3636	3643	3731*	3732*	3733	
3801*	3802*	3808	3810	3817	3904*	3905*	3906	3974*	3975*	3981	3983	3990	
4074*	4075*	4076	4096*	4097*	4098	4113*	4114*	4115	4139*	4140*	4141	4156*	
4157*	4158	4182*	4183*	4189	4191	4198	4281*	4282*	4283	4303*	4304*	4305	
4320*	4321*	4322	4346*	4347*	4348	4363*	4364*	4365	4389*	4390*	4396	4398	
4405	4493*	4494*	4495	4517*	4518*	4524	4526	4533	4620*	4621*	4622	4644*	
4645*	4651	4653	4660	4752*	4753*	4754	4769*	4770*	4771	4799*	4800*	4806	
4808	4815	4906*	4907*	4908	4923*	4924	4925	4953*	4954*	4960	4962	4969	
5076*	5077*	5078	5103*	5108	5111	5133*	5140	5140	5142	5149	5186*	5187*	
5188	5213*	5216	5221	5243*	5244*	5250	5252	5259	5325*	5326*	5327	5342*	
5343*	5349	5351	5358	5461*	5466	5469	5482*	5483*	5484	5499*	5500*	5501	
5527*	5528*	5534	5536	5543	5643*	5648	5651	5664*	5665*	5666	5681*	5682*	
5683	5709*	5710*	5716	5718	5725	5809*	5810*	5811	5829*	5830*	5831	5869*	
5870*	5871	5889*	5890*	5891	5917*	5918*	5924	5926	5933	5968*	5969*	5970	
6037*	6038*	6039	6057*	6058*	6059	6097*	6098*	6099	6117*	6118*	6119	6145*	
6146*	6152	6154	6161	6196*	6197*	6198	6269*	6270*	6271	6289*	6290*	6291	
6329*	6330*	6331	6350*	6351*	6352	6378*	6379*	6385	6387	6394	6430*	6431*	
6432	6502*	6503*	6504	6522*	6523*	6524	6562*	6563*	6564	6583*	6584*	6585	
6611*	6612*	6618	6620	6627	6663*	6664*	6665	6728*	6729*	6730	6748*	6749*	
6750	6773*	6774*	6780	6782	6789	6829*	6830*	6831	6846*	6847*	6848	6911*	
6912*	6913	6931*	6932*	6933	6955*	6956*	6962	6964	6971	7077*	7078*	7079	
7099*	7100*	7106	7108	7115	7193*	7194*	7195	7215*	7216*	7222	7224	7231	
7322*	7323*	7324	7348*	7349*	7355	7357	7364	7397*	7398*	7399	7472*	7473*	
7474	7498*	7499*	7505	7507	7514	7547*	7548*	7549	7635*	7636*	7637	7650*	
7651*	7652	7671*	7672*	7673	7686*	7687*	7688	7715*	7716*	7722	7724	7731	
7760*	7761*	7762	7775*	7776*	7777	7806*	7807*	7813	7815	7822	7902*	7903*	

CZRMGB0 RM03-2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 228
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0227

\$XTSTR 066234
\$\$GET4= 000000
\$OFILL 057425
\$4OCAT= ***** U
= 101014

9629#													
9537#													
9862*	9866*	9876	9911#										
9626	9697												
1451#	1455#	1463	1464#	1466#	1468#	1485#	1530	1971	1984	1985	2070	2071	
2171	2192#	2495	2500	2505#	2510#	2515#	2520	2525	2530	2535	2540	2545	
2723	2728	2733	2738	2743	2748	2753	2758	2763	2768	2773	2892	3059	
3384	3578	3742	3915	5062	5820	5840	6048	6068	6280	6300	6513	6533	
6739	6759	6812	6922	6942	6994	7312	7462	8659	8798	8890	8896	9036	
9042	9507	9522#	9545	9546#	9668	9669	9710	9765#	9836	9979#	9983	9987#	
9988	9989#	10211#	10212	10218#	10272	11078#							

LL	9136#	9332													
LL1	9136#	9255	9451												
MM	7288#	7438													
MMO	6700#														
MM1	5781#	6009													
MORETA	1478#	1531													
MSG	2120#	2125	2320#	2323	2548#	2551	2776#	2779	2943#	2946	3111#	3114	3195#	3198	3280#
	3283	3474#	3477	3669#	3672	3842#	3845	4016#	4019	4223#	4226	4431#	4434	4558#	4561
	4686#	4689	4840#	4843	4995#	4998	5285#	5288	5384#	5387	5566#	5569	5750#	5753	5979#
	5982	6209#	6212	6442#	6445	6674#	6677	6859#	6862	7024#	7027	7140#	7143	7259#	7262
	7410#	7413	7561#	7563	7571#	7573	7581#	7583	7848#	7850	8114#	8117	8318#	8321	8523#
	8526	8662#	8665	8803#	8806	8952#	8955	9099#	9102	9295#	9298				
MULT	1249#														
NEUTRA	1115#	2438	2666	2843	3010	3144	3228	3423	3617	3791	3964	4172	4379	4507	4634
	4789	4943	5123	5233	5332	5517	5699	5907	6135	6368	6601	6763	6945	7089	7205
	7338	7488	7705	7796	7972	8063	8239	8443	8604	8743	8901	9047	9206	9402	9402
NEWST	1249#	2123	2321	2549	2777	2944	3112	3196	3281	3475	3670	3843	4017	4224	4432
	4559	4687	4841	4996	5286	5385	5567	5751	5980	6210	6443	6675	6860	7025	7141
	7260	7411	7581	7848	8115	8319	8524	8663	8804	8953	9100	9296	9492		
NN	8836#	8982													
OO	4047#	4254													
POP	1249#	9966	10261	10310											
PUSH	1249#	9925	10235	10290											
RELEAS	1115#	2837	3004	3388	3417	3582	3611	3785	3958	4166	4373	4501	4628	4783	4937
	5088	5117	5198	5227	5446	5511	5628	5693	5901	6129	6362	6595	7082	7198	7332
	7482	7699	7790	7966	8057	9171	9200	9367	9396						
REPORT	1249#														
RR	7049#	7165													
SCOPE	1144#	2318	2547	2775	2942	3109	3194	3278	3473	3667	3841	4014	4222	4429	4557
	4684	4839	4993	5283	5382	5565	5747	5978	6206	6440	6673	6856	7022	7139	7255
	7409	7559	7845	8112	8317	8521	8661	8800	8951	9097	9254	9450	9493		
SEIZE	1115#	2362	2590	2811	2978	3324	3518	3713	3886	4049	4256	4468	4595	4727	4881
	5428	5610	5844	6072	6304	6537	7618	7885	8152	8356	8564	8703	8849	8995	9147
	9343														
SELECT	1115#	2152	2173	2198	2217	2240	2272	2368	2379	2397	2596	2607	2625	2840	2894
	2911	2934	3007	3061	3078	3101	3132	3216	3330	3341	3359	3391	3401	3404	3420
	3524	3535	3553	3585	3595	3598	3614	3745	3753	3766	3788	3918	3926	3939	3961
	4055	4066	4089	4106	4132	4149	4169	4262	4273	4296	4313	4339	4356	4376	4474
	4485	4504	4601	4612	4631	4733	4744	4758	4786	4887	4898	4912	4940	5044	5052
	5058	5091	5101	5104	5120	5178	5201	5211	5214	5230	5435	5449	5459	5462	5475
	5492	5514	5617	5631	5641	5644	5657	5674	5696	5802	5822	5850	5861	5904	5961
	6030	6050	6078	6089	6132	6189	6262	6282	6310	6321	6365	6423	6495	6515	6543
	6554	6598	6656	6721	6741	6822	6839	6904	6924	7014	7018	7072	7086	7188	7202
	7299	7335	7386	7449	7485	7536	7702	7753	7793	7969	8020	8060	8170	8181	8293
	8374	8385	8497	8594	8733	8856	8863	9002	9009	9156	9159	9164	9168	9174	9184
	9187	9203	9352	9355	9360	9364	9370	9380	9383	9399					
SETATA	1115#	5782	6010	6242	6475	6701	6886								
SETPRI	1249#	10140													
SETTRA	10340#	10349	10350	10351	10352	10354	10356	10357	10358	10359	10360	10361			
SETUP	1249#	1966													
SKIP	1249#														
SLASH	1249#														
SPACE	1249#														
STAR#	1249#	1250	1254	1296	1300	1423	1427	1461	1481	1530	1563	1567	1956	1960	2114
	2118	2123	2135	2149	2195	2237	2305	2321	2338	2355	2361	2394	2411	2427	2434
	2487	2549	2566	2583	2589	2622	2639	2655	2662	2715	2777	2790	2803	2809	2836

CZRMGB0 RMO3/2 DU POR LGC 1
CZRMGB.P11 21-NOV-77 13:24

MACY11 30(1046) 21-NOV-77 13:53 PAGE 233
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0231

.\$SCOP	1115#	9611
.\$STRAP	1115#	10317
.\$TYPD	1115#	9913
.\$TYPE	1115#	9766
.\$TYP0	1115#	9836

. ABS. 101014 000

ERRORS DETECTED: 0

RMO3:CZRMGB.BIN,RMO3:CZRMGB.SEQ/DOC/NL:TOC/SOL/CRF=RMO3:CZRMGB.P11

RUN-TIME: 38 40 3 SECONDS

RUN-TIME RATIO: 1289/83=15.3

CORE USED: 32k (63 PAGES)

DOCUMENT PAGES: 231