

RP04/6

DUAL CONTROLLER LOGIC CZRJEBO

AH-9201B-MC

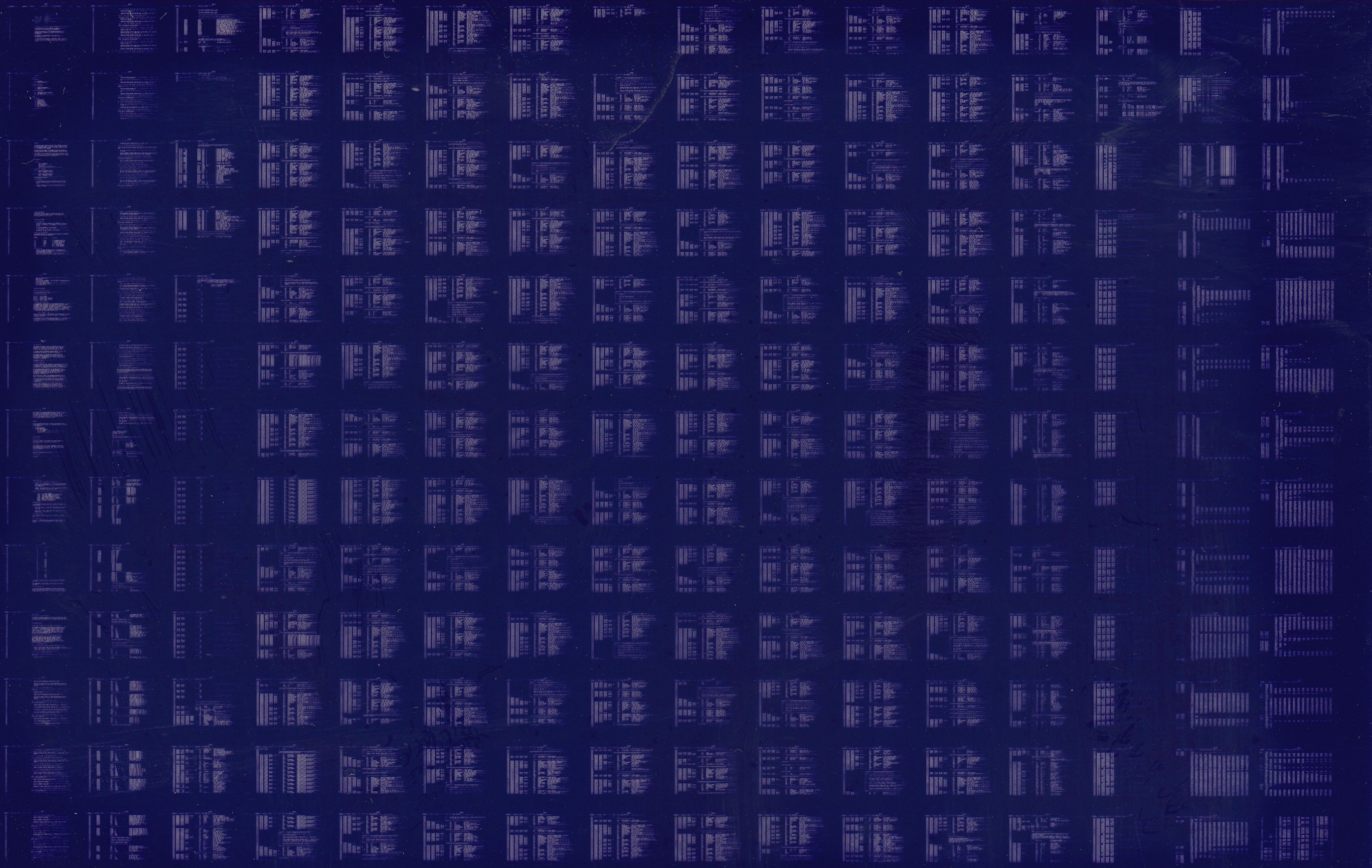
JAN 1978

COPYRIGHT © 74-77

digital

FICHE 1 OF 2

MADE IN USA



RP04/6

**DUAL CONTROLLER LOGIC
CZRJEBO**

AH-9201B-MC

COPYRIGHT © 74-77

FICHE 2 OF 2

JAN 1978

digital

MADE IN USA

89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

1. ABSTRACT

THE RPO4/5/6 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RPO4/5/6 DUAL CONTROLLER LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL CONTROLLER MODE IS NOT TESTED BY THIS PROGRAM.

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 PROCESSOR
16K OF MEMORY
KW11-L OR KW11-P CLOCK
TELETYPE
RH11 OR RH70 WITH AN RPO4/5/6
RPO4/5/6 DUAL CONTROLLER OPTION TEST CABLE

2.2 PRELIMINARY PROGRAMS

RPO4/5/6 DISKLESS CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJG)
PART 2 (MAINDEC-11-DZRJH)

RPO4/5/6 FUNCTIONAL CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJI)
PART 2 (MAINDEC-11-DZRJJ)

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

2.3 OTHER PROGRAMS

A. THE OPERATION OF THE UNLOAD COMMAND AND THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH ARE TESTED BY THE RPO4/5/6 DUAL CONTROLLER LOGIC TEST, PART 2 (MAINDEC-11-DZRJF).

B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RPO4/5/6 MULTIDRIVE EXERCISER PROGRAM (MAINDEC-11-DZRJD).

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

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.

4. STARTING PROCEDURES

4.1 STARTING ADDRESSES

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

4.2 UNIBUS & VECTOR ADDRESSES

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

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

4.3 OPERATOR ACTION

- A. CONNECT THE DUAL CONTROLLER TEST CABLE BETWEEN BUS A & BUS B ON THE DRIVE BEING TESTED. (SEE SECTION 5.4)
- B. LOAD THE PROGRAM INTO MEMORY IN THE PROCESSOR CONTROLLING THE MASSBUS USED FOR TESTING.
- C. SWITCH THE 'CONTROLLER SELECT' SWITCH ON THE DRIVE TO BE TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
- D. LOAD THE APPROPRIATE STARTING ADDRESS (200(8), 204(8) OR 210(8)).

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

- INTO THE SWITCH REGISTER (OR THE 'SOFTWARE' SWITCH REGISTER, REFER TO SECTION 5.2).
- E. PRESS START.
- F. ENTER THE DRIVE NUMBER. (THIS MUST BE THE NUMBER DISPLAYED BY THE DRIVE, IF AN RPO4, OR THE NUMBER OF THE ADDRESS PLUG IF THE DRIVE IS AN RPO5/6).
- G. ENTER THE NUMBER OF THE TEST TO BE RUN. ('CARRIAGE RETURN' OR '0' WILL RUN ALL TESTS.)
- H. THE PROGRAM MAY BE STOPPED AT ANY TIME AND RESTARTED FROM LOCATION 200.

5. OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

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

THE SWITCH SETTINGS ARE:

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

5.2 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RPO4/5/6 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

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

5.3 TEST SELECTION

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

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

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

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

5.4 TEST CABLE CONNECTION

TO TEST THE RPO4/5/6 DUAL CONTROLLER OPTION WITH THIS PROGRAM, A SPECIAL TEST CABLE MUST BE USED. (THE TEST CABLE IS P/N 7010507-02). THE TEST CABLE CONNECTS MASSBUS A & MASSBUS B TOGETHER AT THE DRIVE BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE DRIVE CABLE CONNECTED TO THE RPO4 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT OF THE DRIVE WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS. THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S ADDRESS (THE ADDRESS SELECTED BY THE SWITCHES ON THE 'DP' BOARD - MODULE M7775 FOR RPO4'S, OR BY THE ADDRESS PLUG FOR RPO5/6'S.)

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

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

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

WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION

313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360

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

6. ERRORS

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

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

7. MISCELLANEOUS

7.1 RESTRICTIONS

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

7.2 LIMITATIONS

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

7.3 EXECUTION TIME

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

7.4 STACK POINTER

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

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

7.5 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

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

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

- TYPE - TTY TYPEOUT ROUTINE
- TYPOC - TYPE OCTAL NUMBER (WITH LEADING ZERO)
- TYPOS - TYPE OCTAL NUMBER (NO LEADING ZEROS)
- TYPON - TYPE OCTAL NUMBER PER LAST CALL
- TYPDS - TYPE DECIMAL NUMBER WITH SIGN
- RDCHR - READ CHARACTER FROM TTY KEYBOARD
- RDLIN - READ A LINE FROM THE TTY KEYBOARD.
- RDOCT - READ AN OCTAL NUMBER FROM THE TTY KEYBOARD
- SAVREG - ROUTINE TO SAVE R0-R5
- RESREG - ROUTINE TO RESTORE R0-R5

7.6 REQUIRED TESTS

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

- A. TEST 2 AND TEST 3. THESE TESTS DETERMINE AND STORE FOR LATER USE THE TIMEOUT NON-SHOT VALUE MEASURED THROUGH EACH PORT.
- B. TEST 4 AND TEST 5. THESE TESTS SET 'VV-A' AND 'VV-B' RESPECTIVLY. THESE TESTS MUST BE PERFORMED AT LEASE ONCE BEFORE TESTS 6 - 46 ARE RUN.

7.7 DISK SURFACE USAGE

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

7.8 TEST ITERATIONS

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

ITERATION COUNT

175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

TEST NO.	(IN DECIMAL)
01	1
02	10
03	10
04	1
05	1
06	4000
07	4000
10	100
11	100
12	4000
13	f
14	f
15	f
16	40
17	4000
20	40000
21	40000
22	40000
23	40000
24	40000
25	40000
26	40000
27	40000
28	40000
29	40000
30	40000
31	40000
32	40000
33	40000
34	40000
35	40000
36	40000
37	40000
38	40000
39	40000
40	40000
41	40000
42	40000
43	40000
44	40000
45	40000
46	200

IF AN ERROR OCCURS IN A TEST, THAT TEST WILL BE PERFORMED ONLY ONCE. THE OCCURENCE OF AN ERROR FORCES THE ITERATION COUNT TO '1'.

TEST PERFORMED IN THE SINGLE TEST MODE WILL BE ITERATED THE NUMBER OF TIMES SPECIFIED BY THE ITERATION COUNT FOR THE TEST.

7.9 LOOP ON ERROR OPTION

IF SW<09> IS SET, THE PROGRAM WILL LOOP ON A FAILING TEST UNTIL EITHER THE SWITCH IS RESET OR THE ERROR STOPS OCCURING. BECAUSE THE PROGRAM MUST RESET THE RPO4 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

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

WAS DETECTED.

7.10 SPECIAL M7775 'DP' BOARD TESTS

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

8. TEST DESCRIPTIONS

8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

THE PROGRAM DETERMINES THAT THE DRIVE IS IN NEUTRAL BY CHECKING THE CONTENTS OF THE DRIVE STATUS REGISTER (RPDS1) THROUGH BOTH PORTS. THE PROGRAM MASKS OUT THE PORT DEPENDENT BITS ('ATA' & 'VV') AND VERIFIES THAT CORRECT STATUS IS READ THROUGH BOTH PORTS. (THE CORRECT STATUS IS 'MOL', 'PGM', 'DPR', & 'DRY'.) IF NEITHER PORT SEES ALL ZEROS FROM RPDS1, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DESCREPANCY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

THE PROGRAM VERIFIES THAT THE DRIVE HAS BEEN SEIZED BY CHECKING THE DRIVE STATUS REGISTER (RPDS1) THROUGH THE SEIZING PORT AND VERIFING THAT CORRECT STATUS IS SEEN. WHEN RPDS1 IS READ THROUGH THE OPPOSITE PORT, ZEROS SHOULD BE SEEN. IF BOTH CONDITIONS EXIST, (I.E., CORRECT STATUS THROUGH THE SEIZING PORT AND ZEROS THROUGH THE OPPOSITE PORT), THE PROGRAM CONCLUDES THAT THE DRIVE HAS BEEN SEIZED BY THE SPECIFIED PORT.

8.3 TEST 1 - DRIVE ACCESS TEST

VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS

A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RPO4/5/6, THAT THE DRIVE IS ONLINE (RPDS1 HAS 'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.

B. THE TEST IS REPEATED THROUGH BOTH PORTS.

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

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

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

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

- TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
 - A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- 8.9 TEST 7 - TEST RELEASE, DRIVE SEIZED BY PORT 'B'
 - TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
 - A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- 8.10 TEST 10 - TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
 - TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
 - A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
- 8.11 TEST 11 - TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
 - TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
 - A. ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
- 8.12 TEST 12 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
 - VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING PORT TO RELEASE THE DRIVE.
 - A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
 - B. ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
 - C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
 - D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.13 TEST 13 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
 - VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING PORT TO RELEASE THE DRIVE.
 - A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.

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

VERIFY THAT THE DRIVE HAS BEEN SEIZED.

- B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

B.14 TEST 14 - TEST RESET ATTENTION 'A' BY MASSBUS CLEAR

VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- C. ISSUE A MASSBUS CLEAR.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

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

VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- C. ISSUE A MASSBUS CLEAR.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

B.16 TEST 16 - TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL

VERIFY THAT MASSBUS CLEAR DOES NOT RESET ATTENTION BITS WHEN THE DRIVE IS IN NEUTRAL.

- A. SET THE ATTENTION BITS FOR BOTH PORTS.
- B. VERIFY THAT THE DRIVE IS IN NEUTRAL.
- C. ISSUE A MASSBUS INIT. VERIFY THAT NEITHER ATTENTION BIT HAS RESET.

B.17 TEST 17 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'

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

- VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
- A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'A'; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.18 TEST 20 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
- VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
- A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'B'; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.19 TEST 21 - TEST 'PORT REQUEST' FROM PORT 'A'
- VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE DRIVE IS SEIZED BY THE OTHER PORT.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- B. WRITE 0'S INTO RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'A' AND IS NOT SET FOR PORT 'B'.
- D. ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.20 TEST 22 - TEST PORT REQUEST FROM PORT 'B'
- VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE DRIVE IS SEIZED BY THE OTHER PORT.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. WRITE 0'S INTO RPDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'.
- D. ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.21 TEST 23 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
- VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT

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

- REQUEST'.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
 - B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT SET.
 - C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.22 TEST 24 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'
VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT REQUEST'.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
 - B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT SET.
 - C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.23 TEST 25 - TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE IS SEIZED BY THE OTHER PORT.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
 - C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
 - D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'.
 - E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.24 TEST 26 - TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE IS SEIZED BY THE OTHER PORT.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
 - C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
 - D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'.

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

- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.25 TEST 27 - TEST SEIZE BY WRITING ATTENTION BIT
TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER (RPAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER PORT.
- A. WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
- 8.26 TEST 30 - TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO THE DRIVE'S ATTENTION BIT.
- A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
- B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
- 8.27 TEST 31 - TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
- C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
- 8.28 TEST 32 - TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
- C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
- 8.29 TEST 33 - TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR BITS ARE SET IN THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

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

- B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
 - C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.
 - D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
 - E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.30 TEST 34 - TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
- VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR BITS ARE SET IN THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
 - C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.
 - D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
 - E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.31 TEST 35 - TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
- VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.
 - C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
 - D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.33 TEST 37 - TEST PORT 'A' ATTENTION AFTER A COMMAND
- TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A COMMAND.
- A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
 - B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
 - C. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED

TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

- 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
- B.34 TEST 40 - TEST PORT 'B' ATTENTION AFTER A COMMAND
TEST THE OPERATION OF THE PROT A AND PORT B ATTENTION BITS AFTER A COMMAND.
- ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
 - WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
 - RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- B.35 TEST 41 - TEST PORT INTERACTION FROM PORT 'A'
VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 - CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 - RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
 - ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- B.36 TEST 42 - TEST PORT INTERACTION FROM PORT 'B'
VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 - READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 - CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 - WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 - RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS

985 SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
986 SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
987
988 G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
989 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
990
991 8.37 TEST 43 - TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
992
993 VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
994
995 A. SET THE ATTENTION BIT FOR PORT 'A'.
996
997 B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
998
999 C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
1000 FOR THE DRIVE IS SET.
1001
1002 8.38 TEST 44 - TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
1003
1004 VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
1005
1006 A. SET THE ATTENTION BIT FOR PORT 'B'.
1007
1008 B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
1009
1010 C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
1011 FOR THE DRIVE IS SET.
1012
1013 8.39 TEST 45 - TEST NO TIMEOUT THROUGH PORT 'A'
1014
1015 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
1016 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
1017 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
1018 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST
1019 BE SELECTED SEPARATELY.
1020
1021 A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
1022
1023 B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
1024
1025 C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE
1026 HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
1027 SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
1028
1029 D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
1030 BEEN RELEASED.
1031
1032 E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
1033 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
1034
1035 8.40 TEST 46 - TEST NO TIMEOUT THROUGH PORT 'B'
1036
1037 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
1038 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
1039 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
1040 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST

1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096

BE SELECTED SEPARATELY.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT BEEN RELEASED.
- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

9. PROGRAM LISTING

a

```

.TITLE CZRJEBO, DL CTRLR LGC
.*COPYRIGHT (C) 1976,1977
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY C. HESS
.*
.*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
MT= 11                ;;CODE FOR HORIZONTAL TAB
LF= 12                ;;CODE FOR LINE FEED

```

001100

000011
000012

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

.*GENERAL PURPOSE REGISTER DEFINITIONS

1107	000000	R0=	%0	:: GENERAL REGISTER
1108	000001	R1=	%1	:: GENERAL REGISTER
1109	000002	R2=	%2	:: GENERAL REGISTER
1110	000003	R3=	%3	:: GENERAL REGISTER
1111	000004	R4=	%4	:: GENERAL REGISTER
1112	000005	R5=	%5	:: GENERAL REGISTER
1113	000006	R6=	%6	:: GENERAL REGISTER
1114	000007	R7=	%7	:: GENERAL REGISTER
1115	000006	SP=	%6	:: STACK POINTER
1116	000007	PC=	%7	:: PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS

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

.*"SWITCH REGISTER" SWITCH DEFINITIONS

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

```

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

```



```

1209      001000      A17=      1000      ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
1210      002000      PSEL=     2000      ;PORT SELECT (BIT #10)
1211      020000      MCPE=    20000      ;MASSBUS PARITY ERROR (BIT #13)
1212      040000      TRE=     40000      ;TRANSFER ERROR (BIT #14)
1213      100000      SC=     100000     ;SPECIAL CONDITION (BIT #15)
1214
1215      ;WORD COUNT REGISTER (RPWC)
1216      ;(EACH BIT IS CALLED BY BIT NUMBER)
1217
1218      ;BUS ADDRESS REGISTER (RPBA)
1219      ;(EACH BIT IS CALLED BY BIT NUMBER)
1220
1221      ;CONTROL AND STATUS REGISTER 2 (RPCS2)
1222
1223      000001      US1=       1      ;UNIT SELECT (BIT #0)
1224      000002      US2=       2      ;UNIT SELECT (BIT #1)
1225      000004      US4=       4      ;UNIT SELECT (BIT #2)
1226      000010      BAI=      10      ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
1227      000020      PAT=      20      ;MASSBUS PARITY TEST (BIT #4)
1228      000040      CLR=      40      ;CLEAR (BIT #5)
1229      000100      IR=       100      ;INPUT READY (BIT #6)
1230      000200      OR=       200      ;OUTPUT READY (BIT #7)
1231      000400      MPE=      400      ;MASS BUS PARITY ERROR (BIT #8)
1232      001000      MXF=     1000      ;MISSED TRANSFER ERROR (BIT #9)
1233      002000      PGE=     2000      ;PROGRAM ERROR (BIT #10)
1234      004000      NEM=     4000      ;NON EXISTENT MEMORY (BIT #11)
1235      010000      NED=    10000      ;NON EXISTENT DRIVE (BIT #12)
1236      020000      UPE=    20000      ;UNIBUS PARITY ERROR (BIT #13)
1237      040000      WCE=    40000      ;WRITE CHECK ERROR (BIT #14)
1238      100000      DLT=   100000      ;DATA LATE (BIT #15)
1239
1240      ;DATA BUFFER REGISTER (RPDB)
1241      ;(EACH BIT IS CALLED BY BIT NUMBER)
1242
1243      ;*****
1244      ;.SBTTL RP04/5/6 REGISTERS
1245      ;*****
1246
1247      ;CONTROL AND STATUS 1 REGISTER. (#00)
1248
1249
1250
1251      000001      G0=       1      ;GO BIT (BIT #0)
1252      000002      F1=       2      ;FUNCTION CODE BIT #1
1253      000004      F2=       4      ;FUNCTION CODE BIT #2
1254      000010      F3=      10      ;FUNCTION CODE BIT #3
1255      000020      F4=      20      ;FUNCTION CODE BIT #4
1256      000040      F5=      40      ;FUNCTION CODE BIT #5
1257      004000      DVA=     4000      ;DEVICE AVAILABLE (BIT #11)
1258
1259      ;DRIVE STATUS REGISTER (RPDS1) (#01)
1260
1261
1262      ;DF5=      1      DRIVE FORWARD 5"/SEC. (BIT #0)
1263      000002      DFF20=     2      ;DRIVE FORWARD 20"/SEC. (BIT #1)
1264      000004      DIGB=     4      ;DRIVE TO INNER GUARD BAND (BIT #2)

```

1265 000010
1266 000020
1267 000040
1268 000100
1269 000200
1270 000400
1271 001000
1272 002000
1273 004000
1274 010000
1275 020000
1276 040000
1277 100000

GRV= 10
DL64= 20
DE1= 40
VV= 100
DRY= 200
DPR= 400
PGM= 1000
LST= 2000
WRL= 4000
MOL= 10000
PIP= 20000
ERR= 40000
ATA= 100000

; GO REVERSE (BIT #3)
; DIFFERENCE LESS THAN 64 (BIT #4)
; DIFFERENCE EQUALS 1 (BIT #5)
; VOLUME VALID (BIT #6)
; DRIVE READY (BIT #7)
; DRIVE PRESENT (BIT #8)
; PROGRAMABLE (BIT #9)
; LAST SECTOR TRANSFERRED (BIT #10)
; WRITE LOCK (BIT #11)
; MEDIUM ON-LINE (BIT #12)
; POSITIONING OPERATION IN PROGRESS (BIT #13)
; COMPOSITE ERROR (BIT #14)
; ATTENTION ACTIVE (BIT #15)

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

1281 000001
1282 000002
1283 000004
1284 000010
1285 000020
1286 000040
1287 000100
1288 000200
1289 000400
1290 001000
1291 002000
1292 004000
1293 010000
1294 020000
1295 040000
1296 100000

ILF= 1
ILR= 2
RMR= 4
PAR= 10
FER= 20
WCF= 40
ECH= 100
HCE= 200
HCRC= 400
AOE= 1000
IAE= 2000
WLE= 4000
DTE= 10000
OPI= 20000
UNS= 40000
DCK= 100000

; ILLEGAL FUNCTION (BIT #0)
; ILLEGAL REGISTER (BIT #1)
; REGISTER MODIFICATION REFUSED (BIT #2)
; PARITY ERROR (BIT #3)
; FORMAT ERROR (BIT #4)
; WRITE CLOCK FAIL (BIT #5)
; ECC HARD ERROR (BIT #6)
; HEADER COMPARE ERROR (BIT #7)
; HEADER CRC ERROR (BIT #8)
; ADDRESS OVERFLOW ERROR (BIT #9)
; INVALID ADDRESS ERROR (BIT #10)
; WRITE LOCK ERROR (BIT #11)
; DRIVE TIMING ERROR (BIT #12)
; OPERATION INCOMPLETE (BIT #13)
; DRIVE UNSAFE (BIT #14)
; DATA CHECK ERROR (BIT 15)

; MAINTAINABILITY REGISTER (RPMR) (#03)

1300 000001
1301 000002
1302 000004
1303 000010
1304 000020
1305 000040
1306 000200

DMD= 1
MCLK= 2
MINX= 4
MSTCK= 10
MRD= 20
MWR= 40
DTSY= 200

; DIAGINOSTIC MODE (BIT #0)
; MAINTAINABILITY CLOCK (BIT #1)
; MAINTAINABILITY INDEX (BIT #2)
; MAINTAINABILITY SECTOR CLOCK (BIT #3)
; MAINTAINABILITY READ (BIT #4)
; MAINTAINABILITY WRITE (BIT #5)
; MAINTAINABILITY SYNC DETECTED (BIT #7)

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

1310 000001
1311 000002
1312 000004
1313 000010
1314 000020
1315 000040
1316 000100
1317 000200

AT0= 1
AT1= 2
AT2= 4
AT3= 10
AT4= 20
AT5= 40
AT6= 100
AT7= 200

; DEVICE 0 (BIT #0)
; DEVICE 1 (BIT #1)
; DEVICE 2 (BIT #2)
; DEVICE 3 (BIT #3)
; DEVICE 4 (BIT #4)
; DEVICE 5 (BIT #5)
; DEVICE 6 (BIT #6)
; DEVICE 7 (BIT #7)

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

1318
1319
1320

```

1321
1322
1323
1324      000001      DT00=      1      ;DRIVE TYPE REGISTER (RPDT) (#06)
1325      000002      DT01=      2      ;DRIVE TYPE NUMBER BIT 1
1326      000004      DT02=      4      ;DRIVE TYPE NUMBER BIT 2
1327      000010      DT03=     10      ;DRIVE TYPE NUMBER BIT 3
1328      000020      DT04=     20      ;DRIVE TYPE NUMBER BIT 4
1329      000040      DT05=     40      ;DRIVE TYPE NUMBER BIT 5
1330      000100      DT06=    100      ;DRIVE TYPE NUMBER BIT 6
1331      000200      DT07=    200      ;DRIVE TYPE NUMBER BIT 7
1332      000400      DT08=    400      ;DRIVE TYPE NUMBER BIT 8
1333      004000      DRQ=   4000      ;DRIVE REQUEST REQUIRED (BIT #11)
1334      002000      MOH=  20000      ;MOVING HEAD (BIT #13)
1335      040000      TAP=  40000      ;TAPE DRIVE (BIT #14)
1336      100000      NBA= 100000      ;NOT BLOCK ADDRESSED (BIT #15)
1337
1338
1339
1340      000001      EXT1=      1      ;LOOK-AHEAD REGISTER (RPLA) (#07)
1341      000002      EXT2=      2      ;EXTENSION 1 (BIT #0)
1342      000004      EXT4=      4      ;EXTENSION 2 (BIT #1)
1343      000010      EXT10=     10      ;EXTENSION 3 (BIT #2)
1344      000020      EXT20=     20      ;EXTENSION 4 (BIT #3)
1345      000040      EXT40=     40      ;EXTENSION 5 (BIT #4)
1346      000100      SC1=    100      ;EXTENSION 6 (BIT #5)
1347      000200      SC2=    200      ;SECTOR COUNT FIELD 0 (BIT #6)
1348      000400      SC4=    400      ;SECTOR COUNT FIELD 1 (BIT #7)
1349      001000      SC10=  1000      ;SECTOR COUNT FIELD 2 (BIT #8)
1350      002000      SC20=  2000      ;SECTOR COUNT FIELD 3 (BIT #9)
1351      004000      TRK1=  4000      ;SECTOR COUNT FIELD 4 (BIT #10)
1352      010000      TRK2= 10000      ;TRACK FIELD 1 (BIT #11)
1353      020000      TRK4= 20000      ;TRACK FIELD 2 (BIT #12)
1354      040000      TRK10= 40000      ;TRACK FIELD 3 (BIT #13)
1355      100000      TRK20=100000      ;TRACK FIELD 4 (BIT #14)
1356
1357
1358
1359      000001      WCU=      1      ;RPO4 ERROR REGISTER #2 (RPER2) (#10)
1360      000002      CSF=      2      ;WRITE CURRENT UNSAFE (BIT #0)
1361      000004      WSU=      4      ;CURRENT SINK FAILURE (BIT #1)
1362      000010      CSU=     10      ;WRITE SELECT UNSAFE (BIT #2)
1363      000020      MSE=     20      ;CURRENT SWITCH UNSAFE (BIT #3)
1364      000040      TDF=     40      ;MOTOR SEQUENCE ERROR (BIT #4)
1365      000100      TUF=    100      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1366      000200      FEN=    200      ;TRANSITIONS UNSAFE (BIT #6)
1367      000400      WRU=    400      ;FAILSAFE ENABLED (BIT #7)
1368      001000      MHS=   1000      ;WRITE READY UNSAFE (BIT #8)
1369      002000      NHS=   2000      ;MULTIPLE HEAD SELECT (BIT #9)
1370      004000      IXE=   4000      ;NO HEAD SELECTION (BIT #10)
1371      010000      VU30= 10000      ;INDEX ERROR (BIT #11)
1372      020000      PLU=  20000      ;30VOLT UNSAFE (BIT #12)
1373      100000      ACU= 100000      ;PLO UNSAFE (BIT #13)
1374
1375
1376
1375      ;RPO5/6 ERROR REGISTER #02 (RPER2) (#10)

```

1377 000001
 1378 000002
 1379 000004
 1380 000010
 1381 000020
 1382 000040
 1383 000100
 1384 000200
 1385 000400
 1386 001000
 1387 002000
 1388 004000
 1389 020000
 1390
 1391
 1392
 1393 000001
 1394 000002
 1395 000004
 1396 000010
 1397 000020
 1398 000040
 1399 000200
 1400 002000
 1401 004000
 1402 010000
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415 000001
 1416 000002
 1417 000010
 1418 000020
 1419 000040
 1420 000100
 1421 000200
 1422 100000
 1423
 1424
 1425
 1426 000001
 1427 000002
 1428 000040
 1429 000100
 1430 020000
 1431 040000
 1432 100000

WCU= 1
 CSF= 2
 WSU= 4
 CSU= 10
 RAW= 20
 TDF= 40
 TUF= 100
 ABS= 200
 WRU= 400
 MHS= 1000
 NHS= 2000
 IXE= 4000
 PLU= 20000

;WRITE CURRENT UNSAFE (BIT #0)
 ;CURRENT SINK FAILURE (BIT #1)
 ;WRITE SELECT UNSAFE (BIT #2)
 ;CURRENT SWITCH UNSAFE (BIT #3)
 ;READ AND WRITE (BIT #4)
 ;TRANSITIONS DETECTOR FAILURE (BIT #5)
 ;TRANSITIONS UNSAFE (BIT #6)
 ;ABNORMAL STOP (BIT #7)
 ;WRITE READY UNSAFE (BIT #8)
 ;MULTIPLE HEAD SELECT (BIT #9)
 ;NO HEAD SELECTION (BIT #10)
 ;INDEX ERROR (BIT #11)
 ;PLO UNSAFE (BIT #12)

;OFFSET REGISTER (RPOF) (#11)

OF25= 1
 OF50= 2
 OF100= 4
 OF200= 10
 OF400= 20
 OF800= 40
 OFREV= 200
 HCI= 2000
 ECI= 4000
 FMT22= 10000

;OFFSET 25 MICRO INCHES (BIT #0)
 ;OFFSET 50 MICRO INCHES (BIT #1)
 ;OFFSET 100 MICRO INCHES (BIT #2)
 ;OFFSET 200 MICRO INCHES (BIT #3)
 ;OFFSET 400 MICRO INCHES (BIT #4)
 ;OFFSET 800 MICRO INCHES (BIT #5)
 ;OFFSET NEGATIVE (REVERSE) (BIT #5)
 ;HEADER COMPARE INHIBIT (BIT #10)
 ;ERROR CORRECTION CODE INHIBIT (BIT #11)
 ;FORMAT BIT (BIT #12)

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

;CURRENT CYLINDER ADDRESS (RPCC) (#13)
 ;(EACH BIT IS CALLED BY BIT NUMBER)

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

;RPO4 ERROR REGISTER #03 (RPER3) (#15)

PSU= 1
 VUF= 2
 UWR= 10
 PRE= 20
 ACL= 40
 DCL= 100
 SKI= 40000
 OCYL= 100000

;PACK SPEED UNSAFE (BIT #0)
 ;VELOCITY UNSAFE (BIT #1)
 ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
 ;DISK PACK ROTATION ERROR (BIT #4)
 ;AC LOW (BIT #5)
 ;DC LOW (BIT #6)
 ;SEEK INCOMPLETE (BIT #14)
 ;OFF CYLINDER (BIT #15)

;RPO5/6 ERROR REGISTER #03 (RPER3) (#15)

DCU= 1
 WAO= 2
 ACL= 40
 DCL= 100
 OPE= 20000
 SKI= 40000
 OCYL= 100000

;DC UNSAFE (BIT #0)
 ;WRITE AND OFFSET (BIT #1)
 ;AC LOW (BIT #5)
 ;DC LOW (BIT #6)
 ;OPERATOR PLUG ERROR (BIT #13)
 ;SEEK INCOMPLETE (BIT #14)
 ;OFF CYLINDER ERROR (BIT #15)

```

1433
1434
1435 ;ECC POSITION REGISTER (RPEC1) (#16)
1436 ;(EACH BIT IS CALLED BY BIT NUMBER)
1437
1438 ;ECC PATTERN REGISTER (RPEC2) (#17)
1439 ;(EACH BIT IS CALLED BY BIT NUMBER)
1440
1441 ;;*****
1442
1443 .SBTTL DEFINITIONS OF THE RH11/RPO4/5/6 ADDRESS INDEXES
1444
1445 ;;*****
1446
1447 000000 RPCS1=0 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
1448 000002 RPWC=2 ;WORD COUNT REGISTER (NOT A DRIVE REG)
1449 000004 RPBA=4 ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
1450 000006 RPOA=6 ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
1451 000010 RPCS2=10 ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
1452 000012 RPS1=12 ;DRIVE STATUS REGISTER (DRIVE REG 01)
1453 000014 RPER1=14 ;ERROR REGISTER #1 (DRIVE REG. 02)
1454 000016 RPAS=16 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
1455 000020 RPLA=20 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
1456 000022 RPDB=22 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
1457 000024 RPRM=24 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
1458 000026 RPDT=26 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
1459 000030 RPSN=30 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
1460 000032 RPOF=32 ;OFFSET REGISTER (DRIVE REG. 11)
1461 000034 RPCA=34 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
1462 000036 RPCC=36 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
1463 000040 RPER2=40 ;ERROR REGISTER #2 (DRIVE REG. 14)
1464 000042 RPER3=42 ;ERROR REGISTER #3 (DRIVE REG. 15)
1465 000044 RPEC1=44 ;ECC POSITION REGISTER (DRIVE REG. 16)
1466 000046 RPEC2=46 ;ECC PATTERN REGISTER (DRIVE REG. 17)
1467
1468 .SBTTL TRAP CATCHER
1469
1470 000000 .=0
1471 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1472 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1473 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1474
1475 000174 000174 .=174
1476 000176 000000 DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
1477 000176 000000 SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
1478
1479 .SBTTL ACT11 HOOKS
1480
1481 ;;*****
1482 ;HOOKS REQUIRED BY ACT11
1483 000200 $SVPC=. ;SAVE PC
1484 000046 .=46 $ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
1485 000052 .=52 ;;2)SET LOC.52 TO 20000
1486 000052 .WORD 20000 ;;RESTORE PC
1487 000200 .=$SVPC
1488

```

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 28
CZRJEB.P11 04-NOV-77 13:27 STARTING ADDRESS = 200

SEQ 0028

1489
1490
1491
1492
1493
1494
1495
1496
1497

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

```

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

```

.SBTTL COMMON TAGS

```

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

```

. = 1100

```

SCMTAG: .WORD 0
SPASS: .WORD 0
STSTNM: .BYTE 0
SERFLG: .BYTE 0
SICNT: .WORD 0
SLPADR: .WORD 0
SLPERR: .WORD 0
SERTTL: .WORD 0
SITEMB: .BYTE 0
SERMAX: .BYTE 1
SERRPC: .WORD 0
SGDADR: .WORD 0
SBDADR: .WORD 0
SGDDAT: .WORD 0
SBDDAT: .WORD 0
SAUTOB: .BYTE 0
SINTAG: .BYTE 0
SWR: .WORD DSWR
DISPLAY: .WORD DDISP
STKS: 177560
STKB: 177562
STPS: 177564
STPB: 177566
SNUL: .BYTE 0
SFILLS: .BYTE 2
SFILLC: .BYTE 12
STPFLG: .BYTE 0
SREGAD: .WORD 0
SREGO: .WORD 0
STMP0: .WORD 0
STMP1: .WORD 0
STMP2: .WORD 0
STMP3: .WORD 0
STMP4: .WORD 0
STIMES: 0
SESCAPE: 0
SBELL: .ASCIZ <207><377><377>
SQUES: .ASCII /?/
SCRLF: .ASCII <15>
SLF: .ASCIZ <12>
CR = 15
LF = 12
SLKCSR: .WORD 172540
SLKCSB: .WORD 172542

```

```

: START OF COMMON TAGS
: CONTAINS PASS COUNT
: CONTAINS THE TEST NUMBER
: CONTAINS ERROR FLAG
: CONTAINS SUBTEST ITERATION COUNT
: CONTAINS SCOPE LOOP ADDRESS
: CONTAINS SCOPE RETURN FOR ERRORS
: CONTAINS TOTAL ERRORS DETECTED
: CONTAINS ITEM CONTROL BYTE
: CONTAINS MAX. ERRORS PER TEST
: CONTAINS PC OF LAST ERROR INSTRUCTION
: CONTAINS ADDRESS OF 'GOOD' DATA
: CONTAINS ADDRESS OF 'BAD' DATA
: CONTAINS 'GOOD' DATA
: CONTAINS 'BAD' DATA
: RESERVED--NOT TO BE USED
: AUTOMATIC MODE INDICATOR
: INTERRUPT MODE INDICATOR
: ADDRESS OF SWITCH REGISTER
: ADDRESS OF DISPLAY REGISTER
: TTY KBD STATUS
: TTY KBD BUFFER
: TTY PRINTER STATUS REG. ADDRESS
: TTY PRINTER BUFFER REG. ADDRESS
: CONTAINS NULL CHARACTER FOR FILLS
: CONTAINS # OF FILLER CHARACTERS REQUIRED
: INSERT FILL CHARS. AFTER A "LINE FEED"
: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
: CONTAINS THE ADDRESS FROM
: WHICH (SREGO) WAS OBTAINED
: CONTAINS ((SREGAD)+0)
: USER DEFINED
: USER DEFINED
: USER DEFINED
: USER DEFINED
: USER DEFINED
: MAX. NUMBER OF ITERATIONS
: ESCAPE ON ERROR ADDRESS
: CODE FOR BELL
: QUESTION MARK
: CARRIAGE RETURN
: LINE FEED
*****
: ADDR OF KW11-P STATUS REGISTER
: ADDR OF KW11-P COUNTER BUFFER

```

1554 001216 000104
1555 001220 177546
1556 001222 000100
1557 001224 000000
1558 001226 000000
1559 001230 000000
1560 001232 000000
1561 001234 000000
1562 001236 000000
1563 001240 000000
1564 001242 000000
1565 001244 000000
1566 001246 000000
1567 001250 000000
1568 001252 000000
1569 001254 000000
1570 001256 000000
1571 001260 000000
1572 001262 000000
1573 001264 000000
1574 001266 000000
1575 001270 000000
1576 001272 000000
1577 001274 000000
1578 001276 000000

SLPVEC: .WORD 104
SLKS: .WORD 177546
SLLVEC: .WORD 100
PORTA: .WORD 0
PORTB: .WORD 0
PORTC: .WORD 0
ASR1: .WORD 0
PTNBR: .WORD 0
SEIZPT: .WORD 0
OPPR: .WORD 0
TSTNUM: .WORD 0
CKERR: .WORD 0
NOSEIZ: .WORD 0
RELERR: .WORD 0
TIME: .WORD 0
WATCH: .WORD 0
TIMEA: .WORD 0
TIMEAP: .WORD 0
TIMEAM: .WORD 0
TIMEB: .WORD 0
TIMEBP: .WORD 0
TIMEBM: .WORD 0
TIMES: .WORD 0
KYBCTL: .WORD 0
CHGADR: .WORD 0

; ADDR OF KW11-P VECTOR
; ADDR OF KW11-L STATUS REGISTER
; ADDR OF KW11-L VECTOR
; ADDRESS OF PORT A
; ADDRESS OF PORT B
; ADDRESS OF DIFFERENT DRIVE
; ATA-A OR ATA-B = 1
; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
; CONTAINS THE ADDRESS OF THE SEIZING PORT
; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
; NUMBER OF THE CURRENT TEST
; IF -1, A REGISTER MISCOMPARISON OCCURRED
; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
; ELAPSED TIME COUNTER
; WATCH DOG TIMER LOCATION
; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
; PORT A TIMEOUT VALUE + 25%
; PORT A TIMEOUT VALUE - 25%
; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
; PORT A TIMEOUT VALUE + 25%
; PORT B TIME VALUE - 25%
; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
; SINGLE TEST INDICATOR
; CHANGE THE RH11 ADDRESS INDICATOR

1579
1580
1581
1582
1583
1584
1585
1586 001300 176700
1587 001302 000254
1588

;;*****
.SBTTL RH11/RP04/5/6 UNIBUS AND VECTOR ADDRESSES
;;*****
\$RPADR: .WORD 176700 ;RH11/RP04/5/6 UNIBUS ADDRESS
\$RPVEC: .WORD 254 ;RH11 INTERRUPT VECTOR ADDRESS

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

001304

001304 062476
001306 066534
001310 070272
001312 070542

001314 062517
001316 066534
001320 070272
001322 070542

001324 062541
001326 066605
001330 070306
001332 070542

001334 062623
001336 066654
001340 070354
001342 070555

001344 062654
001346 066777
001350 070322
001352 070547

001354 062722
001356 067247
001360 070374

.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

1645	001362	070547	DF5	
1646				
1647				
1648				
1649	001364	063022	EM7	;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1650	001366	067053	DH7	
1651	001370	070354	DT7	
1652	001372	070555	DF7	
1653				
1654				
1655				
1656	001374	063103	EM10	;REGISTER CONTENTS INCORRECT
1657	001376	066777	DH5	
1658	001400	070322	DT5	
1659	001402	070547	DF5	
1660				
1661				
1662				
1663	001404	063133	EM11	;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1664	001406	067176	DH11	
1665	001410	070272	DT1	
1666	001412	070542	DF1	
1667				
1668				
1669				
1670	001414	063217	EM12	;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1671	001416	067743	DH36	
1672	001420	070462	DT37	
1673	001422	070570	DF36	
1674				
1675				
1676				
1677	001424	063267	EM13	; 'VOLUME VALID' BIT NOT SET BY READIN PRESET
1678	001426	067247	DH13	
1679	001430	070374	DT13	
1680	001432	070547	DF5	
1681				
1682				
1683				
1684	001434	063354	EM14	; 'VOLUME VALID' SET ON THE OPPOSITE PORT
1685	001436	067247	DH13	
1686	001440	070374	DT13	
1687	001442	070547	DF5	
1688				
1689				
1690				
1691	001444	063417	EM15	;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1692	001446	067053	DH7	
1693	001450	070354	DT7	
1694	001452	070555	DF7	
1695				
1696				
1697				
1698	001454	063476	EM16	;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1699	001456	067053	DH7	
1700	001460	070354	DT7	

1701	001462	070555	DF7	
1702				
1703				;ERROR 17
1704				
1705	001464	063551	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
1706	001466	067053	DH7	
1707	001470	070354	DT7	
1708	001472	070555	DF7	
1709				
1710				;ERROR 20
1711				
1712	001474	063630	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
1713	001476	067743	DH36	
1714	001500	070462	DT37	
1715	001502	070570	DF36	
1716				
1717				;ERROR 21
1718				
1719	001504	063710	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
1720	001506	067743	DH36	
1721	001510	070462	DT37	
1722	001512	070570	DF36	
1723				
1724				;ERROR 22
1725				
1726	001514	063763	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
1727	001516	067367	DH22	
1728	001520	070412	DT22	
1729	001522	070564	DF31	
1730				
1731				;ERROR 23
1732				
1733	001524	064050	EM23	;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
1734	001526	067465	DH23	
1735	001530	070424	DT23	
1736	001532	070542	DF1	
1737				

1738			;ERROR 24	
1739				
1740	001534	064116	EM24	;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1741	001536	057465	DH23	
1742	001540	070424	DT23	
1743	001542	070542	DF1	
1744				
1745				
1746			;ERROR 25	
1747				
1748	001544	064175	EM25	;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1749	001546	067743	DH36	
1750	001550	070452	DT36	
1751	001552	070570	DF36	
1752				
1753				
1754			;ERROR 26	
1755				
1756	001554	064240	EM26	;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
1757	001556	067367	DH22	
1758	001560	070412	DT22	
1759	001562	070564	DF31	
1760				
1761			;ERROR 27	
1762				
1763	001564	064325	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1764	001566	067053	DH7	
1765	001570	070354	DT7	
1766	001572	070555	DF7	
1767				
1768			;ERROR 30	
1769				
1770	001574	064403	EM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1771	001576	067743	DH36	
1772	001600	070452	DT36	
1773	001602	070570	DF36	
1774				
1775			;ERROR 31	
1776				
1777	001604	064500	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SE
1778	001606	067644	DH31	
1779	001610	070440	DT31	
1780	001612	070564	DF31	
1781				
1782			;ERROR 32	
1783				
1784	001614	064555	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1785	001616	066777	DH5	
1786	001620	070322	DT5	
1787	001622	070547	DF5	
1788				
1789			;ERROR 33	
1790				
1791	001624	064626	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1792	001626	067743	DH36	
1793	001630	070452	DT36	

1794	001632	070570	DF36	
1795				
1796				;ERROR 34
1797				
1798	001634	064730	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1799	001636	067743	DH36	
1800	001640	070452	DT36	
1801	001642	070570	DF36	
1802				
1803				;ERROR 35
1804				
1805	001644	065033	EM35	;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
1806	001646	067743	DH36	
1807	001650	070462	DT37	
1808	001652	070570	DF36	
1809				
1810				;ERROR 36
1811				
1812	001654	065112	EM36	;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
1813	001656	067743	DH36	
1814	001660	070452	DT36	
1815	001662	070570	DF36	
1816				
1817				;ERROR 37
1818				
1819	001664	065164	EM37	;DRIVE IS NON-EXISTENT
1820	001666	067743	DH36	
1821	001670	070462	DT37	
1822	001672	070570	DF36	
1823				
1824				;ERROR 40
1825				
1826	001674	065232	EM40	;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1827	001676	066534	DH1	
1828	001700	070424	DT23	
1829	001702	070542	DF1	
1830				
1831				;ERROR 41
1832				
1833	001704	065307	EM41	;TIMEOUT CLEARED ATTENTION BIT
1834	001706	067465	DH23	
1835	001710	070424	DT23	
1836	001712	070542	DF1	
1837				
1838				;ERROR 42
1839				
1840	001714	065351	EM42	;DRIVE NOT IN NEUTRAL OR SEIZED
1841	001716	067772	DH42	
1842	001720	070472	DT42	
1843	001722	070573	DF42	
1844				
1845				;ERROR 43
1846				
1847	001724	065437	EM43	;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1848	001726	067772	DH42	
1849	001730	070472	DT42	

1850	001732	070573	DF42	
1851				
1852				;ERROR 44
1853				
1854	001734	065514	EM44	;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1855	001736	070011	DH44	
1856	001740	070440	DT31	
1857	001742	070564	DF31	
1858				
1859				;ERROR 45
1860				
1861	001744	065571	EM45	;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1862	001746	066534	DH1	
1863	001750	070272	DT1	
1864	001752	070542	DF1	
1865				
1866				;ERROR 46
1867				
1868	001754	065650	EM46	;CAN'T ACCESS DRIVE THROUGH EITHER PORT
1869	001756	070107	DH46	
1870	001760	070500	DT46	
1871	001762	070564	DF31	
1872				
1873				;ERROR 47
1874				
1875	001764	065717	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
1876	001766	067465	DH23	
1877	001770	070424	DT23	
1878	001772	070542	DF1	
1879				
1880				;ERROR 50
1881				
1882	001774	066005	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
1883	001776	067247	DH13	
1884	002000	070374	DT13	
1885	002002	070547	DF5	
1886				
1887				;ERROR 51
1888				
1889	002004	066070	EM51	;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1890	002006	066777	DH5	
1891	002010	070322	DT5	
1892	002012	070547	DF5	
1893				
1894				;ERROR 52
1895				
1896	002014	066153	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1897	002016	067247	DH13	
1898	002020	070374	DT13	
1899	002022	070547	DF5	
1900				
1901				;ERROR 53
1902				
1903	002024	066251	EM53	;CAN'T READ ATTN BIT FROM OPPOSITE PORT
1904	002026	067465	DH23	
1905	002030	070272	DT1	

```

1906 002032 070542          DF1
1907
1908          ;ERROR 54
1909
1910 002034 066332          EM54          ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1911 002036 067367          DH22
1912 002040 070512          DT54
1913 002042 070564          DF31
1914
1915          ;ERROR 55
1916
1917 002044 066425          EM55          ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1918 002046 070205          DH55
1919 002050 070524          DT55
1920 002052 070575          DF55
1921
1922          ;ERROR 56
1923
1924 002054 066472          EM56          ;RH11 DIDN'T RESPOND TO ADDRESSING
1925 002056 070263          DH56
1926 002060 070536          DT56
1927 002062 070601          DF56
1928
1929
1930
1931          ;;*****
1932          .SBTTL  STARTUP AND INITIALIZATION ROUTINES
1933          ;;*****
1934
1935
1936
1937 002064 005037 001276  START:  CLR      CHGADR          ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
1938 002070 000403          BR      START2          ;GO TO THE START
1939 002072 012737 177777 001276  START1: MOV      #-1,CHGADR      ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
1940 002100 000005          START2: RESET          ;CLEAR THE BUS
1941          .SBTTL  INITIALIZE THE COMMON TAGS
1942          ;;CLEAR THE COMMON TAGS (SCMTAG) AREA
1943 002102 012706 001100          MOV      #SCMTAG,R6          ;;FIRST LOCATION TO BE CLEARED
1944 002106 005026          CLR      (R6)+          ;;CLEAR MEMORY LOCATION
1945 002110 022706 001140          CMP      #SWR,R6 ;;DONE?
1946 002114 001374          BNE     #-6          ;;LOOP BACK IF NO
1947 002116 012706 001100          MOV      #STACK,SP          ;;SETUP THE STACK POINTER
1948          ;;INITIALIZE A FEW VECTORS
1949 002122 012737 056670 000020          MOV      #SCOPE,#IOTVEC      ;;IOT VECTOR FOR SCOPE ROUTINE
1950 002130 012737 000340 000022          MOV      #340,#IOTVEC+2      ;;LEVEL 7
1951 002136 012737 057122 000030          MOV      #ERROR,#EMTVEC      ;;EMT VECTOR FOR ERROR ROUTINE
1952 002144 012737 000340 000032          MOV      #340,#EMTVEC+2      ;;LEVEL 7
1953 002152 012737 061722 000034          MOV      #TRAP,#TRAPVEC      ;;TRAP VECTOR FOR TRAP CALLS
1954 002160 012737 000340 000036          MOV      #340,#TRAPVEC+2      ;;LEVEL 7
1955 002166 013737 056270 056262          MOV      SENDCT,#EOPCT      ;;SETUP END-OF-PROGRAM COUNTER
1956 002174 005037 001176          CLR      $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1957 002200 005037 001200          CLR      $ESCAPE          ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1958 002204 112737 000001 001115          MOV      #1,$ERMAX          ;;ALLOW ONE ERROR PER TEST
1959 002212 012737 002212 001106          MOV      #,$SLPADR          ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1960 002220 012737 002220 001110          MOV      #,$SLPERR          ;;SETUP THE ERROR LOOP ADDRESS
1961          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS

```

M03

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 38
 CZRJEBO.P11 04-NOV-77 13:27

INITIALIZE THE COMMON TAGS

SEQ 0038

```

1962 ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1963 002226 013746 000004 MOV @ERRVEC, -(SP) ;; SAVE ERROR VECTOR
1964 002232 012737 002266 000004 MOV @64$, @ERRVEC ;; SET UP ERROR VECTOR
1965 002240 012737 177570 001140 MOV @DSWR, SWR ;; SETUP FOR A HARDWARE SWICH REGISTER
1966 002246 012737 177570 001142 MOV @DDISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
1967 002254 022777 177777 176656 CMP @-1, @SWR ;; TRY TO REFERENCE HARDWARE SWR
1968 002262 001012 BNE 66$ ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
1969 ;; AND THE HARDWARE SWR IS NOT = -1
1970 002264 000403 BR 65$ ;; BRANCH IF NO TIMEOUT
1971 002266 012716 002274 64$: MOV @65$, (SP) ;; SET UP FOR TRAP RETURN
1972 002272 000002 RTI
1973 002274 012737 000176 001140 65$: MOV @SWREG, SWR ;; POINT TO SOFTWARE SWR
1974 002302 012737 000174 001142 MOV @DISPREG, DISPLAY
1975 002310 012637 000004 66$: MOV (SP)+, @ERRVEC ;; RESTORE ERROR VECTOR
1976
1977 002314 005227 177777 INC @-1 ;; FIRST START ?
1978 002320 001002 BNE 1$ ;; BR IF NOT
1979 002322 104401 062010 TYPE TITLE ;; TYPE PROGRAM NAME
1980 002326 004737 060332 1$: JSR PC, $TKINT ;; SETUP THE TTY KEYBOARD
1981 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
1982 002332 005737 000042 TST @42 ;; ARE WE RUNNING UNDER XXDP/ACT?
1983 002336 001006 BNE 67$ ;; BRANCH IF YES
1984 002340 023727 001140 000176 CMP SWR, @SWREG ;; SOFTWARE SWITCH REG SELECTED?
1985 002346 001005 BNE 68$ ;; BRANCH IF NO
1986 002350 104406 GTSWR ;; GET SOFT-SWR SETTINGS
1987 002352 000403 BR 68$
1988 002354 112737 000001 001134 67$: MOV @1, $AUTOB ;; SET AUTO-MODE INDICATOR
1989 002362 002362 68$:
1990 002362 004737 002754 JSR PC, CHANGE ;; CHECK/CHANGE THE RH11 ADDRESS
1991 002366 104401 062107 TYPE ,ENTERA ;; ENTER DRIVE ADDRESS
1992 002372 104412 RDOCT ;; GET THE ADDRESS
1993 002374 012637 001224 MOV (SP)+, PORTA ;; STORE THE ADDRESS
1994 002400 023727 001224 000007 CMP PORTA, @7 ;; SEE IF ADDRESS TOO LARGE
1995 002406 101403 BLOS 2$ ;; BR IF NOT
1996 002410 104401 062137 TYPE ,ADRERR ;; TYPE ADDRESS ERROR MESSAGE
1997 002414 000744 BR 1$ ;; TRY AGAIN
1998 002416 013737 001224 001226 2$: MOV PORTA, PORTB ;; GENERATE THE PORT B ADDRESS
1999 002424 005237 001226 INC PORTB ;; INCREMENT THE ADDRESS
2000 002430 042737 000016 001226 BIC @16, PORTB ;; LEAVE BIT 0
2001 002436 013746 001224 MOV PORTA, -(SP) ;; PUT PORT A ADDRESS ON THE STACK
2002 002442 042716 177771 BIC @16, (SP) ;; SAVE BITS 1 & 2
2003 002446 052637 001226 BIS (SP)+, PORTB ;; SET BITS 1 & 2 IN PORT B ADDRESS
2004 002452 104401 062161 TYPE ,PORTAIS ;; 'PORT A ADDRESS IS '
2005 002456 013746 001224 MOV PORTA, -(SP) ;; SAVE PORTA FOR TYPEOUT
2006 ;; TYPE PORT A ADDRESS
2007 002462 104403 TYPOS ;; GO TYPE--OCTAL ASCII
2008 002464 001 .BYTE 1 ;; TYPE 1 DIGIT(S)
2009 002465 000 .BYTE 0 ;; SUPPRESS LEADING ZEROS
2010 002466 104401 062207 TYPE ,PORTBIS ;; 'PORT B ADDRESS IS '
2011 002472 013746 001226 MOV PORTB, -(SP) ;; SAVE PORTB FOR TYPEOUT
2012 ;; TYPE PORT B ADDRESS
2013 002476 104403 TYPOS ;; GO TYPE--OCTAL ASCII
2014 002500 001 .BYTE 1 ;; TYPE 1 DIGIT(S)
2015 002501 000 .BYTE 0 ;; SUPPRESS LEADING ZEROS
2016 002502 104401 001207 TYPE ,SCRLF ;; ANOTHER CR-LF
2017 002506 013737 001224 001230 MOV PORTA, PORTC ;; GENERATE ADDRESS OF DRIVE NOT TESTED
  
```



```

2018 002514 062737 000006 001230 ADD #6,PORTC ;COMPLEMENT SOME BITS
2019 002522 042737 177770 001230 BIC #C7,PORTC ;SAVE ONLY LOWER BITS
2020 002530 013701 001224 MOV PORTA,R1 ;USE PORT A ADDRESS AS INDEX
2021 002534 116137 070716 001232 MOVB ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
2022 002542 005037 001256 CLR TIMEA ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2023 002546 005037 001260 CLR TIMEAP ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2024 002552 005037 001264 CLR TIMEB ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2025 002556 005037 001266 CLR TIMEBP ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2026 002562 004737 056444 JSR PC,CKCLK ;SETUP CLOCK
2027 002566 000137 002602 JMP EXEC ;CLOCK HAS BEEN STARTED
2028 002572 104401 062235 TYPE ,NOCLOCK ;NO CLOCK ON SYSTEM
2029 002576 000000 3$: HALT ;FATAL ERROR
2030 002600 000776 BR 3$ ;INTERLOCK THE HALT

;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
EXEC: RESET ;CLEAR EVERYTHING
CLR PS ;CLEAR THE PROCESSOR STATUS WORD
TYPE $CRLF ;CR-LF
MOV $RPADR,RO ;RH11 ADDRESS FOR INDEXING
MOV #STACK,SP ;LOAD STACK POINTER
JSR PC,CKCLK ;START THE CLOCK
NOP ;RETURN IF NO CLOCK
JSR PC,STKINT ;INITIALIZE THE KEYBOARD
CLR KYBCTL ;CLEAR SINGLE TEST INDICATOR
CLR $PASS ;CLEAR THE PASS COUNT
MOVB #1,$ERMAX ;SET ERROR MAX TO 1
MOV #,$SLPADR ;INITIAL SETTING FOR LOOP ADDRESS
MOV #,$SLPERR ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
1$: TYPE ,TESTNO ;ASK FOR TEST NUMBER
RDOCT ;GET THE NUMBER
MOV (SP)+,R1 ;PUT ENTRY INTO R1
BNE 2$ ;BR IF NOT ZERO
JMP TST1 ;ENTER ZERO - PERFORM ALL TESTS
2$: CMP R1,MAXTN ;SEE IF NUMBER GREATER THAN MAXIMUM
BLE 3$ ;BR IF LESS OR EQUAL
TYPE ,BADNO ;BAD ENTRY
BR 1$ ;TRY AGAIN
3$: DEC R1 ;DECREMENT ENTRY
ASL R1 ;SHIFT IT LEFT
MOV TSTADR(R1),4$ ;GET THE TEST ADDRESS
INC KYBCTL ;SET SINGLE TEST INDICATOR
MOV #1,$ICNT ;PRESET ITERATION COUNT
JMP 24$ ;GO TO THE SELECTED TEST
4$: .WORD 0 ;TEST ADDRESS GOES HERE

;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
CHANGE: TST CHGADR ;CHANGE THE ADDRESS ?
BEQ 3$ ;BR IF NOT
CLR CHGADR ;CLEAR THE INDICATOR
1$: TYPE ,ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS
MOV $RPADR,-(SP) ;PUT THE ADDRESS ON THE STACK
TYPOC ;TYPE THE ACTUAL ADDRESS
TYPE ,SCRLF ;CR-LF
TYPE ,NTRH11 ;ASK FOR NEW ADDRESS
    
```

```

2074 003010 104412 RDOCT
2075 003012 005716 TST (SP) ;D OR 'CR' ENTERED ?
2076 003014 001402 BEQ 2$ ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
2077 003016 011637 001300 MOV (SP),SRPADR ;NEW RH11 ADDRESS
2078 003022 005726 2$: TST (SP) ;CORRECT THE STACK POINTER
2079 003024 012737 003044 000004 3$: MOV #4$,$#4 ;LOAD TRAP ADDRESS
2080 003032 013700 001300 MOV SRPADR,RO ;RH11 ADDRESS
2081 003036 005760 000002 TST RPMC(RO) ;SEE IF RH11 RESPONDS AT THAT ADDRESS
2082 003042 000404 BR 5$ ;BR, RH11 ALIVE AT PRESENT ADDRESS
2083 003044 104056 4$: ERROR 56 ;NO RESPONSE TO ADDRESS
2084 003046 062706 000004 ADD #4,SP ;RESET THE STACK POINTER
2085 003052 000745 BR 1$ ;GET ADDRESS AGAIN
2086 003054 012737 000006 000004 5$: MOV #6,$#4 ;RESTORE THE VECTOR
2087 003062 000207 RTS PC ;RETURN

```

;;*****

.SBTTL *** TESTS ***

;;*****

```

2096 003064 013700 001300 TST1AA: MOV SRPADR,RO ;;RESTORE RO AFTER END OF PASS

```

;;*****

```

2099 *TEST 1 DRIVE ACCESS TEST
2100 *
2101 *VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
2102 *
2103 * A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
2104 * DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RPDS1 HAS
2105 * 'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL
2106 * NUMBER READ THROUGH BOTH PORTS IS THE SAME.
2107 *
2108 * B. THE TEST IS REPEATED THROUGH BOTH PORTS.
2109 *

```

;;*****

```

2111 003070 TST1: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2112 003070 005737 001274 BEQ 2$ ;BR IF NOT
2113 003074 001406 BPL 1$ ;BR IF JUST ENTERED TEST
2114 003076 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2115 003100 000137 002602 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2116 003104 012737 177777 001274 2$: MOV #1,$STNM ;TEST NUMBER
2117 003112 112737 000001 001102 MOV #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
2118 003120 012737 003142 001106 MOV #TEST1,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2119 003126 012737 003142 001110 MOV #1,$TIMES ;DO 1 ITERATION
2120 003134 012737 000001 001176 TEST1: MOV #STACK,SP ;LOAD THE STACK POINTER
2121 003142 012706 001100

```

;;*****

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

```

2126 003146 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A
2127 003154 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2128 003162 005760 000012 TST RPDS1(RO) ;SEE IF DRIVE (PORT A) PRESENT
2129 003166 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR

```

```

2130 003172 016037 000010 001126 MOV RPCS2(R0),SBDDAT ;GET CONTENTS OF RPCS2
2131 003200 012737 000010 001122 MOV #RPCS2,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2132 003206 060037 001122 ADD R0,SBADR ;ADD RH11 BASE ADDRESS
2133 003212 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
2134 003216 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2135 003224 042737 167777 001164 BIC #1CNEB,$TMP0 ;SAVE SPECIFIED BITS
2136 003232 023737 001124 001164 CMP SGDDAT,$TMP0 ;COMPARE THE BITS
2137 003240 001414 BEQ 64$ ;BR IF OK
2138 003242 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
2139 003250 042737 010000 001174 BIC #NEB,$TMP4 ;CLEAR THE MASKED BITS
2140 003256 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2141 003264 104037 ERROR 37 ;TYPE MESSAGE 37
2142 003266 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2143 003272 000240 64$: NOP
2144 003274 005737 001244 TST CKERR ;WAS 'NED' SET ?
2145 003300 001403 BEQ .+10 ;BR IF NOT
2146 003302 012760 000040 000010 MOV #CLR,RPCS2(R0) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2147 003310 113760 001226 000010 MOV#B PORTB,RPCS2(R0) ;SELECT PORT B
2148 003316 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2149 003324 005760 000012 TST RPO51(R0) ;SEE IF DRIVE (PORT B) PRESENT
2150 003330 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2151 003334 016037 000010 001126 MOV RPCS2(R0),SBDDAT ;GET CONTENTS OF RPCS2
2152 003342 012737 000010 001122 MOV #RPCS2,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2153 003350 060037 001122 ADD R0,SBADR ;ADD RH11 BASE ADDRESS
2154 003354 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
2155 003360 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2156 003366 042737 167777 001164 BIC #1CNEB,$TMP0 ;SAVE SPECIFIED BITS
2157 003374 023737 001124 001164 CMP SGDDAT,$TMP0 ;COMPARE THE BITS
2158 003402 001414 BEQ 66$ ;BR IF OK
2159 003404 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
2160 003412 042737 010000 001174 BIC #NEB,$TMP4 ;CLEAR THE MASKED BITS
2161 003420 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2162 003426 104037 ERROR 37 ;TYPE MESSAGE 37
2163 003430 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2164 003434 000240 66$: NOP
2165 003436 005737 001244 TST CKERR ;WAS 'NED' SET ?
2166 003442 001403 BEQ .+10 ;BR IF NOT
2167 003444 012760 000040 000010 MOV #CLR,RPCS2(R0) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2168
2169 ;*****
2170 ;CONFIRM THAT DRIVE IS AN RPO4/5/6 AND IS DUAL PORT
2171
2172 003452 113760 001224 000010 MOV#B PORTA,RPCS2(R0) ;SELECT PORT A
2173 003460 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2174 003466 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2175 003472 016037 000026 001126 MOV RPD1(R0),SBDDAT ;GET CONTENTS OF RPD1
2176 003500 012737 000026 001122 MOV #RPD1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2177 003506 060037 001122 ADD R0,SBADR ;ADD RH11 BASE ADDRESS
2178 003512 012737 024020 001124 MOV #24020,$GDDAT ;WHAT REGISTER SHOULD BE
2179 003520 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2180 003526 042737 000003 001164 BIC #1C17774,$TMP0 ;SAVE SPECIFIED BITS
2181 003534 023737 001124 001164 CMP SGDDAT,$TMP0 ;COMPARE THE BITS
2182 003542 001414 BEQ 68$ ;BR IF OK
2183 003544 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
2184 003552 042737 177774 001174 BIC #17774,$TMP4 ;CLEAR THE MASKED BITS
2185 003560 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

```

2186 003566 104001          ERROR 1          ;TYPE MESSAGE 1
2187 003570 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2188 003574 000240          NOP
2189 003576 113760 001226 000010 68$:  MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
2190 003604 013737 001226 001234    MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2191 003612 005037 001244          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
2192 003616 016037 000026 001126    MOV     RPDT(RO),SBDDAT ;GET CONTENTS OF RPDT
2193 003624 012737 000026 001122    MOV     #RPDT,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2194 003632 060037 001122          ADD     RO,SBADR ;ADD RH11 BASE ADDRESS
2195 003636 012737 024020 001124    MOV     #24020,SGDDAT ;WHAT REGISTER SHOULD BE
2196 003644 013737 001126 001164    MOV     SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2197 003652 042737 000003 001164    BIC     #177774,STMP0 ;SAVE SPECIFIED BITS
2198 003660 023737 001124 001164    CMP     SGDDAT,STMP0 ;COMPARE THE BITS
2199 003666 001414          BEQ     70$          ;BR IF OK
2200 003670 013737 001126 001174    MOV     SBDDAT,STMP4 ;COPY 'BAD DATA'
2201 003676 042737 177774 001174    BIC     #177774,STMP4 ;CLEAR THE MASKED BITS
2202 003704 053737 001174 001124    BIS     STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2203 003712 104001          ERROR 1          ;TYPE MESSAGE 1
2204 003714 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2205 003720 000240          NOP
2206
2207 ;*****
2208 ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
2209
2210 003722 113760 001224 000010  MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
2211 003730 013737 001224 001234    MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2212 003736 005037 001244          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
2213 003742 016037 000012 001126    MOV     RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2214 003750 012737 000012 001122    MOV     #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2215 003756 060037 001122          ADD     RO,SBADR ;ADD RH11 BASE ADDRESS
2216 003762 012737 001000 001124    MOV     #PGM,SGDDAT ;WHAT REGISTER SHOULD BE
2217 003770 013737 001126 001164    MOV     SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2218 003776 042737 176777 001164    BIC     #176777,STMP0 ;SAVE SPECIFIED BITS
2219 004004 023737 001124 001164    CMP     SGDDAT,STMP0 ;COMPARE THE BITS
2220 004012 001414          BEQ     72$          ;BR IF OK
2221 004014 013737 001126 001174    MOV     SBDDAT,STMP4 ;COPY 'BAD DATA'
2222 004022 042737 001000 001174    BIC     #PGM,STMP4 ;CLEAR THE MASKED BITS
2223 004030 053737 001174 001124    BIS     STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2224 004036 104045          ERROR 45         ;TYPE MESSAGE 45
2225 004040 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2226 004044 000240          NOP
2227 004046 005037 001244          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
2228 004052 016037 000012 001126    MOV     RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2229 004060 012737 000012 001122    MOV     #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2230 004066 060037 001122          ADD     RO,SBADR ;ADD RH11 BASE ADDRESS
2231 004072 012737 010600 001124    MOV     #MOL!DPR!DRY,SGDDAT ;WHAT REGISTER SHOULD BE
2232 004100 013737 001126 001164    MOV     SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2233 004106 042737 167177 001164    BIC     #10600,STMP0 ;SAVE SPECIFIED BITS
2234 004114 023737 001124 001164    CMP     SGDDAT,STMP0 ;COMPARE THE BITS
2235 004122 001414          BEQ     74$          ;BR IF OK
2236 004124 013737 001126 001174    MOV     SBDDAT,STMP4 ;COPY 'BAD DATA'
2237 004132 042737 010600 001174    BIC     #10600,STMP4 ;CLEAR THE MASKED BITS
2238 004140 053737 001174 001124    BIS     STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2239 004146 104002          ERROR 2          ;TYPE MESSAGE 2
2240 004150 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2241 004154 000240          NOP

```

E04

```

2242 004156 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
2243 004164 013737 001226 001234      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2244 004172 005037 001244      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2245 004176 016037 000012 001126      MOV   RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2246 004204 012737 000012 001122      MOV   #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2247 004212 060037 001122      ADD   RO,SBADR ;ADD RH11 BASE ADDRESS
2248 004216 012737 001000 001124      MOV   #PGM,SGDDAT ;WHAT REGISTER SHOULD BE
2249 004224 013737 001126 001164      MOV   SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2250 004232 042737 176777 001164      BIC   #1CPGM,$TMP0 ;SAVE SPECIFIED BITS
2251 004240 023737 001124 001164      CMP   SGDDAT,$TMP0 ;COMPARE THE BITS
2252 004246 001414      BEQ   76$ ;BR IF OK
2253 004250 013737 001126 001174      MOV   SBDDAT,$TMP4 ;COPY 'BAD DATA'
2254 004256 042737 001000 001174      BIC   #PGM,$TMP4 ;CLEAR THE MASKED BITS
2255 004264 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2256 004272 104045      ERROR 45 ;TYPE MESSAGE 45
2257 004274 005137 001244      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2258 004300 000240      NOP   76$:
2259 004302 005037 001244      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2260 004306 016037 000012 001126      MOV   RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2261 004314 012737 000012 001122      MOV   #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2262 004322 060037 001122      ADD   RO,SBADR ;ADD RH11 BASE ADDRESS
2263 004326 012737 010600 001124      MOV   #MOL!DPR!DRY,SGDDAT ;WHAT REGISTER SHOULD BE
2264 004334 013737 001126 001164      MOV   SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2265 004342 042737 167177 001164      BIC   #1C10600,$TMP0 ;SAVE SPECIFIED BITS
2266 004350 023737 001124 001164      CMP   SGDDAT,$TMP0 ;COMPARE THE BITS
2267 004356 001414      BEQ   78$ ;BR IF OK
2268 004360 013737 001126 001174      MOV   SBDDAT,$TMP4 ;COPY 'BAD DATA'
2269 004366 042737 010600 001174      BIC   #10600,$TMP4 ;CLEAR THE MASKED BITS
2270 004374 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2271 004402 104002      ERROR 2 ;TYPE MESSAGE 2
2272 004404 005137 001244      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2273 004410 000240      NOP   78$:
2274
2275
2276 ;*****
2277 ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
2278
2279 004412 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A
2280 004420 016037 000030 001124      MOV   RPSN(RO),SGDDAT ;STORE THE PORT A SERIAL NUMBER
2281 004426 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B
2282 004434 016037 000030 001126      MOV   RPSN(RO),SBDDAT ;STORE THE PORT B SERIAL NUMBER
2283 004442 023737 001124 001126      CMP   SGDDAT,SBDDAT ;ARE THEY THE SAME ?
2284 004450 001406      BEQ   1$ ;BR IF THEY ARE
2285 004452 104003      ERROR 3 ;REPORT THE ERROR
2286 004454 032777 100000 174456      BIT   #SW15,JSWR ;HALT ON ERROR ?
2287 004462 001001      BNE   1$ ;BR IF SET - PROGRAM HAS ALREADY HALTED
2288 004464 000000      HALT ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
2289 004466 000004      SCOPE 1$:
2290
2291
2292 ;*****
2293 ;*TEST 2 PORT 'A' SEIZE/TIMEOUT TEST
2294
2295 ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
2296 ;* THE DRIVE.
2297 ;* A. WRITE 0'S INTO RPDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE

```

```

2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309 004470
2310 004470 005737 001274
2311 004474 001406
2312 004476 100002
2313 004500 000137 002602
2314 004504 012737 177777 001274
2315 004512 112737 000002 001102
2316 004520 012737 004542 001106
2317 004526 012737 004542 001110
2318 004534 012737 000012 001176
2319 004542 012706 001100
2320 004546 012737 000240 177776
2321 004554 005037 001256
2322 004560 005037 001260
2323 004564 005037 001262
2324
2325
2326
2327
2328 004570 005037 001252
2329 004574 012737 003720 001254
2330
2331
2332
2333
2334
2335 004602 113760 001224 000010
2336 004610 013737 001224 001236
2337 004616 005060 000012
2338 004622 113760 001226 000010
2339 004630 013737 001226 001234
2340 004636 013737 001226 001240
2341 004644 016037 000012 001126
2342 004652 010037 001122
2343 004656 062737 000012 001122
2344 004664 005037 001124
2345 004670 023737 001124 001126
2346 004676 001403
2347 004700 104004
2348 004702 000137 006066
2349 004706
2350 004706 113760 001224 000010
2351 004714 013737 001224 001234
2352 004722 016037 000012 001126
2353 004730 012737 011600 001124

```

```

; * HAS BEEN SEIZED.
; *
; * B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'B';
; * VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
; *
; * C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
; * MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
; * VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
; * NEUTRAL.
; *
; *****
; TST2:
; 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 #2, $TSTNM ; TEST NUMBER
; MOV #TEST2, $LPADR ; LOAD LOOP ON TEST ADDRESS
; MOV #TEST2, $LPERR ; LOAD LOOP ON ERROR ADDRESS
; MOV #10, $TIMES ; DO 10. ITERATIONS
TEST2: MOV #STACK, SP ; LOAD THE STACK POINTER
; MOV #(<5*32.>), $#PS ; SET PRIORITY TO 5 IN CASE LOOPING
; CLR TIMEA ; CLEAR TIMEOUT VALUE FOR PORT A
; CLR TIMEAP ; CLEAR UPPER TIMEOUT TOLERANCE
; CLR TIMEAM ; CLEAR LOWER TIMEOUT TOLERANCE
; *****
; START THE TIMER
; CLR TIME ; CLEAR THE ELAPSED TIME COUNTER
; MOV #2000., WATCH ; SET WATCH TO 2000 MS
; *****
; SEIZE THE DRIVE THROUGH PORT A
; MOVB PORTA, RPCS2(RO) ; SELECT PORT A
; MOV PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
; CLR RPDS1(RO) ; WRITE RPDS1
; MOVB PORTB, RPCS2(RO) ; SELECT PORT B
; MOV PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
; MOV PORTB, OPPRT ; 'OPPOSITE' PORT ADDRESS
; MOV RPDS1(RO), $BDDAT ; SEE IF DRIVE SEIZED BY PORT A
; MOV RO, $BDADR ; R#11 BASE ADDRESS
; ADD #RPDS1, $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 5$ ; BYPASS REST OF THE SUBTEST
64$: MOVB PORTA, RPCS2(RO) ; SELECT PORT A
; MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
; MOV RPDS1(RO), $BDDAT ; SEE IF SEIZING PORT SEES CORRECT STATUS
; MOV #MOL!PGM!DPR!DRY, $GDDAT ; EXPECTED STATUS

```

```

2354 004736 013737 001124 001166      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
2355 004744 005137 001166      COM      $TMP1 ;COMPLEMENT THE EXPECTED STATUS
2356 004750 013737 001126 001164      MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
2357 004756 043737 001166 001164      BIC      $TMP1,$TMP0 ;CLEAR UNWANTED BITS
2358 004764 023737 001124 001164      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
2359 004772 001401      BEQ      65$ ;BR IF THEY ARE
2360 004774 104005      ERROR 5 ;REPORT THE ERROR
2361 004776 000240      65$: NOP
2362
2363 ;*****
2364 ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2365
2366 005000 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
2367 005006 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2368 005014 016046 000046      MOV      RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT B, FOR CHECK
2369 005020 016046 000044      MOV      RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT B, FOR CHECK
2370 005024 016046 000042      MOV      RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT B, FOR CHECK
2371 005030 016046 000030      MOV      RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT B, FOR CHECK
2372 005034 016046 000036      MOV      RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT B, FOR CHECK
2373 005040 016046 000034      MOV      RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT B, FOR CHECK
2374 005044 016046 000032      MOV      RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT B, FOR CHECK
2375 005050 016046 000040      MOV      RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT B, FOR CHECK
2376 005054 016046 000020      MOV      RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT B, FOR CHECK
2377 005060 016046 000026      MOV      RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT B, FOR CHECK
2378 005064 016046 000006      MOV      RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT B, FOR CHECK
2379 005070 016046 000024      MOV      RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT B, FOR CHECK
2380 005074 016046 000014      MOV      RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT B, FOR CHECK
2381
2382 ;*****
2383 ;WAIT FOR PORT A TO TIMEOUT
2384
2385 005100 005760 000012      1$: TST      RPDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
2386 005104 001006      BNE      2$ ;BR WHEN TIMEOUT OCCURS
2387 005106 005737 001254      TST      WATCH ;CHECK WATCH
2388 005112 001372      BNE      1$ ;BR IF NOT ZERO
2389 005114 104036      ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
2390 005116 000137 005506      JMP      4$ ;BYPASS TIMEOUT TIME CHECK
2391 005122 012737 000340 177776      2$: MOV      #(<7*32.>),D#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2392 005130 013737 001252 001256      MOV      TIME,TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
2393 005136 004537 056630      JSR      RS,TOLER ;CALCULATE THE TOLERANCE
2394 005142 001256      .WORD   TIMEA ;TIMEOUT VALUE FOR PORT A
2395 005144 012637 001260      MOV      (SP)+,TIMEAP ;+25% TOLERANCE
2396 005150 012637 001262      MOV      (SP)+,TIMEAM ;-25% TOLERANCE
2397
2398 ;*****
2399 ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2400
2401 005154 023727 001252 000764      CMP      TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
2402 005162 103001      BHS     3$ ;BR IF IT WAS
2403 005164 104055      ERROR 55 ;REPORT TIMEOUT TOO SHORT
2404
2405 ;*****
2406 ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2407
2408 005166 012737 000240 177776      3$: MOV      #(<5*32.>),D#PS ;RESTORE PRIORITY TO 5
2409

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

2410
2411
2412 005174 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2413 005200 012737 000012 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2414 005206 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
2415 005212 012737 011600 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2416 005220 113760 001224 MOVB PORTA,PCS2(RO) ;SELECT PORT A.
2417 005226 016037 000012 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2418 005234 013737 001170 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
2419 005242 042737 100100 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2420 005250 113760 001226 MOVB PORTB,PCS2(RO) ;SELECT PORT B.
2421 005256 016037 000012 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2422 005264 013737 001172 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
2423 005272 042737 100100 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2424 005300 023737 001164 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2425 005306 001006 BNE 66$ ;BR IF NOT
2426 005310 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2427 005314 001037 BNE 68$ ;BR IF NOT
2428 005316 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2429 005320 000137 JMP 70$ ;BYPASS THE REST OF THE CHECKS
2430 005324 013737 001170 MOV $TMP2,$BDDAT 66$: ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2431 005332 013737 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2432 005340 113760 001226 MOVB PORTB,PCS2(RO) ;SELECT PORT B.
2433 005346 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2434 005352 001414 BEQ 67$ ;BR IF ZERO
2435 005354 013737 001224 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2436 005362 013737 001172 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2437 005370 113760 001224 MOVB PORTA,PCS2(RO) ;SELECT PORT A.
2438 005376 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2439 005402 001004 BNE 68$ ;BR IF NOT
2440 005404 012737 177777 MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
2441 005412 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
2442 005414 013737 001170 MOV $TMP2,$BDDAT 68$: ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
2443 005422 013737 001224 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
2444 005430 042737 100100 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
2445 005436 023737 001124 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
2446 005444 001401 BEQ 69$ ;BR IF OK FROM PORT A.
2447 005446 104007 ERROR 7 ;REPORT ERROR
2448 005450 013737 001172 MOV $TMP3,$BDDAT 69$: ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2449 005456 013737 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2450 005464 042737 100100 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
2451 005472 023737 001124 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
2452 005500 001401 BEQ 70$ ;BR IF OK
2453 005502 104007 ERROR 7 ;REPORT ERROR
2454 005504 000240 70$: NOP

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

2460 005506 013737 001226 MOV PORTB,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2461 005514 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RPER1
2462 005520 062737 000014 ADD #RPER1,$BDADR ;ADDRESS OF RPER1 FOR TYPEOUT
2463 005526 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
2464 005532 001401 BEQ .+4 ;CONTENTS ZERO ?
2465 005534 104006 ERROR 6 ;REPORT THAT PORT B SAW NON-ZERO REGISTER

```


2466	005536	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPMR
2467	005542	062737	000024	001122	ADD	#RPMR, \$BDADR	; ADDRESS OF RPMR FOR TYPEOUT
2468	005550	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPMR
2469	005554	001401			BEQ	.+4	; CONTENTS ZERO ?
2470	005556	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2471	005560	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPDA
2472	005564	062737	000006	001122	ADD	#RPDA, \$BDADR	; ADDRESS OF RPDA FOR TYPEOUT
2473	005572	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPDA
2474	005576	001401			BEQ	.+4	; CONTENTS ZERO ?
2475	005600	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2476	005602	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPDT
2477	005606	062737	000026	001122	ADD	#RPDT, \$BDADR	; ADDRESS OF RPDT FOR TYPEOUT
2478	005614	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPDT
2479	005620	001401			BEQ	.+4	; CONTENTS ZERO ?
2480	005622	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2481	005624	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPLA
2482	005630	062737	000020	001122	ADD	#RPLA, \$BDADR	; ADDRESS OF RPLA FOR TYPEOUT
2483	005636	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPLA
2484	005642	001401			BEQ	.+4	; CONTENTS ZERO ?
2485	005644	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2486	005646	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPER2
2487	005652	062737	000040	001122	ADD	#RPER2, \$BDADR	; ADDRESS OF RPER2 FOR TYPEOUT
2488	005660	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPER2
2489	005664	001401			BEQ	.+4	; CONTENTS ZERO ?
2490	005666	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2491	005670	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPOF
2492	005674	062737	000032	001122	ADD	#RPOF, \$BDADR	; ADDRESS OF RPOF FOR TYPEOUT
2493	005702	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPOF
2494	005706	001401			BEQ	.+4	; CONTENTS ZERO ?
2495	005710	104006			ERROR	6	; REPORT THAT PORT B SAW NON-ZERO REGISTER
2496	005712	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPCA
2497	005716	062737	000034	001122	ADD	#RPCA, \$BDADR	; ADDRESS OF RPCA FOR TYPEOUT
2498	005724	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPCA
2499	005730	001401			BEQ	.+4	; CONTENTS ZERO ?
2500	005732	104006			ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2501	005734	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPCC
2502	005740	062737	000036	001122	ADD	#RPCC, \$BDADR	; ADDRESS OF RPCC FOR TYPEOUT
2503	005746	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPCC
2504	005752	001401			BEQ	.+4	; CONTENTS ZERO ?
2505	005754	104006			ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2506	005756	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPSN
2507	005762	062737	000030	001122	ADD	#RPSN, \$BDADR	; ADDRESS OF RPSN FOR TYPEOUT
2508	005770	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPSN
2509	005774	001401			BEQ	.+4	; CONTENTS ZERO ?
2510	005776	104006			ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2511	006000	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPER3
2512	006004	062737	000042	001122	ADD	#RPER3, \$BDADR	; ADDRESS OF RPER3 FOR TYPEOUT
2513	006012	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPER3
2514	006016	001401			BEQ	.+4	; CONTENTS ZERO ?
2515	006020	104006			ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2516	006022	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPEC1
2517	006026	062737	000044	001122	ADD	#RPEC1, \$BDADR	; ADDRESS OF RPEC1 FOR TYPEOUT
2518	006034	012637	001126		MOV	(SP)+, \$BDDAT	; CHECK THE STORED CONTENTS OF RPEC1
2519	006040	001401			BEQ	.+4	; CONTENTS ZERO ?
2520	006042	104006			ERROR	6	; REPORT THAT PORT B SEES NON-ZERO REGISTER
2521	006044	010037	001122		MOV	RO, \$BDADR	; BASE ADDRESS FOR REGISTER RPEC2

2522 006050 062737 000046 001122
2523 006056 012637 001126
2524 006062 001401
2525 006064 104006
2526 006066 000004

ADD #RPEC2,\$BDADR ;ADDRESS OF RPEC2 FOR TYPEOUT
MOV (SP)+,\$BDDAT ;CHECK THE STORED CONTENTS OF RPEC2
BEQ +4 ;CONTENTS ZERO ?
ERROR 6 ;REPORT THAT PORT B SEES NON-ZERO REGISTER
SS: SCOPE ;LOOP ?

2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544

;TEST 3 PORT 'B' SEIZE/TIMEOUT TEST
;*****
;VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
;THE DRIVE.
;A. WRITE 0'S INTO RPDS1 THROUGH PORT 'B'; VERIFY THAT THE DRIVE
;HAS BEEN SEIZED.
;B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'A';
;VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
;C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
;MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
;VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
;NEUTRAL.
;*****

2545
2546 006070
2547 006070 005737 001274
2548 006074 001406
2549 006076 100002
2550 006100 000137 002602
2551 006104 012737 177777 001274
2552 006112 112737 000003 001102
2553 006120 012737 006142 001106
2554 006126 012737 006142 001110
2555 006134 012737 000012 001176
2556 006142 012706 001100
2557 006146 012737 000240 177776
2558 006154 005037 001264
2559 006160 005037 001266
2560 006164 005037 001270

TST3: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2\$;BR IF NOT
BPL 1\$;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1\$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2\$: MOV #3,\$STNM ;TEST NUMBER
MOV #TEST3,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST3,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #10,\$TIMES ;DO 10. ITERATIONS
TEST3: MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #(<5*32.>),2#PS ;SET PRIORITY TO 5 IN CASE LOOPING
CLR TIMEB ;CLEAR TIMEOUT VALUE FOR PORT B
CLR TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
CLR TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE

2561
2562
2563
2564
2565 006170 005037 001252
2566 006174 012737 003720 001254

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

2567
2568
2569
2570
2571
2572 006202 113760 001226 000010
2573 006210 013737 001226 001236
2574 006216 005060 000012
2575 006222 113760 001224 000010
2576 006230 013737 001224 001234
2577 006236 013737 001224 001240

;SEIZE THE DRIVE THROUGH PORT B
MOV #PORTB,RPCS2(R0) ;SELECT PORT B
MOV #PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR #RPDS1(R0) ;WRITE RPDS1
MOV #PORTA,RPCS2(R0) ;SELECT PORT A
MOV #PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

```

2578 006244 016037 000012 001126      MOV      RPDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
2579 006252 010037 001122      MOV      RO,SBADR ;R#11 BASE ADDRESS
2580 006256 062737 000012 001122      ADD      #RPDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
2581 006264 005037 001124      CLR      $GDDAT ;REGISTER SHOULD BE ZERO
2582 006270 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
2583 006276 001403      BEQ      64$ ;BR IF IT IS
2584 006300 104004      ERROR   4 ;REPORT THE ERROR
2585 006302 000137 007466      JMP      5$ ;BYPASS REST OF THE SUBTEST
2586 006306
2587 006306 113760 001226 000010 64$: MOV      PORTB,RPCS2(RO) ;SELECT PORT B
2588 006314 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2589 006322 016037 000012 001126      MOV      RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2590 006330 012737 011600 001124      MOV      #MOL!PGM!OPR!DRY,$GDDAT ;EXPECTED STATUS
2591 006336 013737 001124 001166      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
2592 006344 005137 001166      COM      $TMP1 ;COMPLEMENT THE EXPECTED STATUS
2593 006350 013737 001126 001164      MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
2594 006356 043737 001166 001164      BIC      $TMP1,$TMP0 ;CLEAR UNWANTED BITS
2595 006364 023737 001124 001164      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
2596 006372 001401      BEQ      65$ ;BR IF THEY ARE
2597 006374 104005      ERROR   5 ;REPORT THE ERROR
2598 006376 000240 65$: NOP
2599
2600 ;*****
2601 ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2602
2603 006400 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
2604 006406 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2605 006414 016046 000046      MOV      RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
2606 006420 016046 000044      MOV      RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
2607 006424 016046 000042      MOV      RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
2608 006430 016046 000030      MOV      RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT A, FOR CHECK
2609 006434 016046 000036      MOV      RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT A, FOR CHECK
2610 006440 016046 000034      MOV      RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT A, FOR CHECK
2611 006444 016046 000032      MOV      RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT A, FOR CHECK
2612 006450 016046 000040      MOV      RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
2613 006454 016046 000020      MOV      RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT A, FOR CHECK
2614 006460 016046 000026      MOV      RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT A, FOR CHECK
2615 006464 016046 000006      MOV      RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT A, FOR CHECK
2616 006470 016046 000024      MOV      RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT A, FOR CHECK
2617 006474 016046 000014      MOV      RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK
2618
2619 ;*****
2620 ;WAIT FOR PORT B TO TIMEOUT
2621
2622 006500 005760 000012 1$: TST      RPDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
2623 006504 001006      BNE      2$ ;BR WHEN TIMEOUT OCCURS
2624 006506 005737 001254      TST      WATCH ;CHECK WATCH
2625 006512 001372      BNE      1$ ;BR IF NOT ZERO
2626 006514 104036      ERROR   36 ;NO TIMEOUT WITHIN 2 SECONDS
2627 006516 000137 007106      JMP      4$ ;BYPASS TIMEOUT TIME CHECK
2628 006522 012737 000340 177776 2$: MOV      #(<7*32.>),#PS ;SET PRIORITY TO 7 TO STOP CLOCK
2629 006530 013737 001252 001264      MOV      TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
2630 006536 004537 056630      JSR      RS,TOLER ;CALCULATE THE TOLERANCE
2631 006542 001264      .WORD   TIMEB ;TIMEOUT VALUE FOR PORT B
2632 006544 012637 001266      MOV      (SP)+,TIMEBP ;+25% TOLERANCE
2633 006550 012637 001270      MOV      (SP)+,TIMEBM ;-25% TOLERANCE

```

2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689

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

006554 023727 001252 000764 CMP TIME,#500 ;WAS MEASURED TIME AT LEAST 500 MS?
006562 103001 BHIS 35 ;BR IF IT WAS
006564 104055 ERROR 55 ;REPORT TIMEOUT TOO SHORT

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

006566 012737 000240 177776 3S: MOV #(<5*32.>),@#PS ;RESTORE PRIORITY TO 5
;VERIFY THAT THE DRIVE IS IN NEUTRAL

006574 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
006600 012737 000012 001122 MOV #RPDS1,\$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
006606 060037 001122 ADD RO,\$BDADR ;ADD THE I/O BASE ADDRESS
006612 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,\$GDDAT ;COMPARISON CONSTANT
006620 113760 001224 000010 MOV#B PORTA,RPDS2(RO) ;SELECT PORT A.
006626 016037 000012 001170 MOV RPDS1(RO),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
006634 013737 001170 001164 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
006642 042737 100100 001164 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006650 113760 001226 000010 MOV#B PORTB,RPDS2(RO) ;SELECT PORT B.
006656 016037 000012 001172 MOV RPDS1(RO),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
006664 013737 001172 001166 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
006672 042737 100100 001166 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006700 023737 001164 001166 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
006706 001006 BNE 66S ;BR IF NOT
006710 005737 001164 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
006714 001037 BNE 68S ;BR IF NOT
006716 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
006720 000137 007104 JMP 70S ;BYPASS THE REST OF THE CHECKS
006724 013737 001170 001126 66S: MOV \$TMP2,\$BDADR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
006732 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006740 113760 001226 000010 MOV#B PORTB,RPDS2(RO) ;SELECT PORT B.
006746 005737 001164 TST \$TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
006752 001414 BEQ 67S ;BR IF ZERO
006754 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006762 013737 001172 001126 MOV \$TMP3,\$BDADR ;'BAD DATA' FOR ERROR TYPE OUT
006770 113760 001224 000010 MOV#B PORTA,RPDS2(RO) ;SELECT PORT A.
006776 005737 001166 TST \$TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
007002 001004 BNE 68S ;BR IF NOT
007004 012737 177777 001250 67S: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
007012 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
007014 013737 001170 001126 68S: MOV \$TMP2,\$BDADR ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
007022 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
007030 042737 100100 001170 BIC #ATA!VV,\$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
007036 023737 001124 001170 CMP \$GDDAT,\$TMP2 ;ALL BITS OK ?
007044 001401 BEQ 69S ;BR IF OK FROM PORT A.
007046 104007 ERROR 7 ;REPORT ERROR
007050 013737 001172 001126 69S: MOV \$TMP3,\$BDADR ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
007056 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
007064 042737 100100 001172 BIC #ATA!VV,\$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
007072 023737 001124 001172 CMP \$GDDAT,\$TMP3 ;SEE IF READ OK FROM PORT B.
007100 001401 BEQ 70S ;BR IF OK

2690 007102 104007
2691 007104 000240

70S: ERROR 7 ;REPORT ERROR
NOP

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

2697	007106	013737	001224	001234	4S:	MOV	PORTA,PTNBR	:CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2698	007114	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPER1
2699	007120	062737	000014	001122		ADD	#RPER1,\$BDADR	:ADDRESS OF RPER1 FOR TYPEOUT
2700	007126	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPER1
2701	007132	001401				BEQ	.+4	:CONTENTS ZERO ?
2702	007134	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2703	007136	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPMR
2704	007142	062737	000024	001122		ADD	#RPMR,\$BDADR	:ADDRESS OF RPMR FOR TYPEOUT
2705	007150	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPMR
2706	007154	001401				BEQ	.+4	:CONTENTS ZERO ?
2707	007156	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2708	007160	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPDA
2709	007164	062737	000006	001122		ADD	#RPDA,\$BDADR	:ADDRESS OF RPDA FOR TYPEOUT
2710	007172	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPDA
2711	007176	001401				BEQ	.+4	:CONTENTS ZERO ?
2712	007200	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2713	007202	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPDT
2714	007206	062737	000026	001122		ADD	#RPDT,\$BDADR	:ADDRESS OF RPDT FOR TYPEOUT
2715	007214	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPDT
2716	007220	001401				BEQ	.+4	:CONTENTS ZERO ?
2717	007222	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2718	007224	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPLA
2719	007230	062737	000020	001122		ADD	#RPLA,\$BDADR	:ADDRESS OF RPLA FOR TYPEOUT
2720	007236	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPLA
2721	007242	001401				BEQ	.+4	:CONTENTS ZERO ?
2722	007244	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2723	007246	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPER2
2724	007252	062737	000040	001122		ADD	#RPER2,\$BDADR	:ADDRESS OF RPER2 FOR TYPEOUT
2725	007260	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPER2
2726	007264	001401				BEQ	.+4	:CONTENTS ZERO ?
2727	007266	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2728	007270	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPOF
2729	007274	062737	000032	001122		ADD	#RPOF,\$BDADR	:ADDRESS OF RPOF FOR TYPEOUT
2730	007302	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPOF
2731	007306	001401				BEQ	.+4	:CONTENTS ZERO ?
2732	007310	104006				ERROR	6	:REPORT THAT PORT A SAW NON-ZERO REGISTER
2733	007312	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPCA
2734	007316	062737	000034	001122		ADD	#RPCA,\$BDADR	:ADDRESS OF RPCA FOR TYPEOUT
2735	007324	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPCA
2736	007330	001401				BEQ	.+4	:CONTENTS ZERO ?
2737	007332	104006				ERROR	6	:REPORT THAT PORT A SEES NON-ZERO REGISTER
2738	007334	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPCC
2739	007340	062737	000036	001122		ADD	#RPCC,\$BDADR	:ADDRESS OF RPCC FOR TYPEOUT
2740	007346	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPCC
2741	007352	001401				BEQ	.+4	:CONTENTS ZERO ?
2742	007354	104006				ERROR	6	:REPORT THAT PORT A SEES NON-ZERO REGISTER
2743	007356	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPSN
2744	007362	062737	000030	001122		ADD	#RPSN,\$BDADR	:ADDRESS OF RPSN FOR TYPEOUT
2745	007370	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPSN

```

2746 007374 001401 BEQ .+4 ;CONTENTS ZERO ?
2747 007376 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2748 007400 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RPER3
2749 007404 062737 000042 001122 ADD #RPER3,$BDADR ;ADDRESS OF RPER3 FOR TYPEOUT
2750 007412 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER3
2751 007416 001401 BEQ .+4 ;CONTENTS ZERO ?
2752 007420 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2753 007422 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RPEC1
2754 007426 062737 000044 001122 ADD #RPEC1,$BDADR ;ADDRESS OF RPEC1 FOR TYPEOUT
2755 007434 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC1
2756 007440 001401 BEQ .+4 ;CONTENTS ZERO ?
2757 007442 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2758 007444 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RPEC2
2759 007450 062737 000046 001122 ADD #RPEC2,$BDADR ;ADDRESS OF RPEC2 FOR TYPEOUT
2760 007456 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC2
2761 007462 001401 BEQ .+4 ;CONTENTS ZERO ?
2762 007464 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2763 007466 000004 5S: SCOPE ;LOOP ?

```

```

*****
:TEST 4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
*
*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
* FOR THE PORT UNDER TEST.
*
* A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE
* DRIVE WAS SEIZED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
*
* B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE
* 'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
* FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
* 'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
*
* C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
* THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
*
*****

```

```

2784 007470 005737 001274 1S: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2785 007470 001406 BEQ 2S ;BR IF NOT
2786 007474 001406 BPL 1S ;BR IF JUST ENTERED TEST
2787 007476 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2788 007500 000137 002602 1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2789 007504 012737 177777 2S: MOV #4,$TSTNM ;TEST NUMBER
2790 007512 112737 000004 001102 MOV #TEST4,$LPADR ;LOAD LOOP ON TEST ADDRESS
2791 007520 012737 007542 001106 MOV #TEST4,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2792 007526 012737 007542 001110 MOV #1,$TIMES ;DO 1 ITERATION
2793 007534 012737 000001 001176 TEST4: MOV #STACK,SP ;LOAD THE STACK POINTER
2794 007542 012706 001100 MOV PORTA,APCS2(RO) ;SELECT PORT A
2795 007546 113760 001224 000010 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2796 007554 013737 001224 001234
2797
2798
2799
2800
2801 007562 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER

```

```

*****
:START THE TIMER

```

```

2802 007566 012737 003720 001254 MOV #2000, WATCH ;SET WATCH TO 2000 MS
2803 007574 013737 001224 001236 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
2804
2805 ;*****
2806 ;ISSUE DRIVE CLEAR COMMAND
2807
2808 007602 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE A DRIVE CLEAR
2809
2810 ;*****
2811 ;VERIFY THAT DRIVE SEIZED BY PORT A.
2812
2813 007610 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B
2814 007616 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2815 007624 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2816 007630 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2817 007636 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2818 007644 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
2819 007650 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
2820 007654 023737 001124 001126 CMP $GDDAT,SBDDAT ;IS THE REGISTER OK ?
2821 007662 001403 BEQ 64$ ;BR IF OK
2822 007664 104012 ERROR 12 ;TYPE MESSAGE 12
2823 007666 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2824 007672 000240 64$: NOP
2825 007674 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A
2826 007702 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2827 007710 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2828 007714 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2829 007722 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2830 007730 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
2831 007734 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
2832 007742 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2833 007750 042737 106177 001164 BIC #1C71600,$TMP0 ;SAVE SPECIFIED BITS
2834 007756 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
2835 007764 001414 BEQ 66$ ;BR IF OK
2836 007766 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
2837 007774 042737 071600 001174 BIC #71600,$TMP4 ;CLEAR THE MASKED BITS
2838 010002 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2839 010010 104010 ERROR 10 ;REPORT THE ERROR
2840 010012 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2841 010016 000240 66$: NOP
2842 010020 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2843 010024 016037 000000 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
2844 010032 012737 000000 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2845 010040 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
2846 010044 012737 004210 001124 MOV #4210,$GDDAT ;WHAT REGISTER SHOULD BE
2847 010052 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2848 010060 042737 100000 001164 BIC #1C77777,$TMP0 ;SAVE SPECIFIED BITS
2849 010066 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
2850 010074 001414 BEQ 68$ ;BR IF OK
2851 010076 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
2852 010104 042737 077777 001174 BIC #77777,$TMP4 ;CLEAR THE MASKED BITS
2853 010112 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2854 010120 104010 ERROR 10 ;REPORT THE ERROR
2855 010122 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2856 010126 000240 68$: NOP
2857

```

```

2858 ;*****
2859 ;ISSUE READIN PRESET COMMAND AND SET FMT22
2860
2861 010130 012760 000023 000000      MOV      #23,RPCS1(RO) ;ISSUE A READIN PRESET
2862 010136 012760 010000 000032      MOV      #FMT22,RPOF(RO) ;SET FMT22
2863
2864 ;*****
2865 ;VERIFY THAT THE DRIVE STATUS IS CORRECT
2866
2867 010144 005037 001244      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
2868 010150 016037 000012 001126      MOV      RPDS1(RO),SDDAT ;GET CONTENTS OF RPDS1
2869 010156 012737 000012 001122      MOV      #RPDS1,SBDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2870 010164 060037 001122      ADD      RO,SBDADR      ;ADD RH11 BASE ADDRESS
2871 010170 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
2872 010176 013737 001126 001164      MOV      SDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO 'STMP0'
2873 010204 042737 106077 001164      BIC      #1C71700,$TMP0 ;SAVE SPECIFIED BITS
2874 010212 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
2875 010220 001414      BEQ      70$           ;BR IF OK
2876 010222 013737 001126 001174      MOV      SDDAT,$TMP4    ;COPY 'BAD DATA'
2877 010230 042737 071700 001174      BIC      #71700,$TMP4   ;CLEAR THE MASKED BITS
2878 010236 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
2879 010244 104013      ERROR   13           ;TYPE MESSAGE 13
2880 010246 005137 001244      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2881 010252 000240      NOP
2882 010254 113760 001226 000010 70$:  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
2883 010262 013737 001226 001234      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2884
2885 ;*****
2886 ;WAIT FOR TIMEOUT TO RELEASE DRIVE
2887
2888 010270 005760 000012      1$:  TST     RPDS1(RO)     ;WAIT FOR THE PORT TO TIME OUT
2889 010274 001006      BNE     2$           ;BR WHEN TIMEOUT OCCURS
2890 010276 005737 001254      TST     WATCH        ;CHECK THE WATCH
2891 010302 001372      BNE     1$          ;BR IF NOT ZERO
2892 010304 104036      ERROR   36          ;NO TIMEOUT WITHIN 2 SECONDS
2893 010306 000137 010624      JMP     3$           ;BYPASS ATTN REGISTER CHECK
2894
2895 ;*****
2896 ;SEE IF DRIVE RETURNED TO NEUTRAL
2897
2898 010312      2$:
2899
2900 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2901
2902 010312 005037 001250      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
2903 010316 012737 000012 001122      MOV      #RPDS1,SBDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2904 010324 060037 001122      ADD      RO,SBDADR     ;ADD THE I/O BASE ADDRESS
2905 010330 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2906 010336 113760 001224 000010      MOVB    PORTA,RPCS2(RO) ;SELECT PORT A.
2907 010344 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2908 010352 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO 'STMP0'
2909 010360 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2910 010366 113760 001226 000010      MOVB    PORTB,RPCS2(RO) ;SELECT PORT B.
2911 010374 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2912 010402 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO 'STMP1'
2913 010410 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```



```

2914 010416 023737 001164 001166      CMP      $TMP0,$TMP1      ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2915 010424 001006                      BNE      72$              ; BR IF NOT
2916 010426 005737 001164              TST      $TMP0            ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
2917 010432 001037                      BNE      74$              ; BR IF NOT
2918 010434 104046                      ERROR    46              ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2919 010436 000137 010622              JMP      76$              ; BYPASS THE REST OF THE CHECKS
2920 010442 013737 001170 001126 72$:    MOV      $TMP2,$BDDAT     ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2921 010450 013737 001226 001234      MOV      PORTB,PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2922 010456 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ; SELECT PORT B.
2923 010464 005737 001164              TST      $TMP0            ; SEE IF STATUS EQ 0 FROM PORT A.
2924 010470 001414                      BEQ      73$              ; BR IF ZERO
2925 010472 013737 001224 001234      MOV      PORTA,PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2926 010500 013737 001172 001126      MOV      $TMP3,$BDDAT     ; 'BAD DATA' FOR ERROR TYPE OUT
2927 010506 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ; SELECT PORT A.
2928 010514 005737 001166              TST      $TMP1            ; SEE IF STATUS EQ ZERO FROM PORT B.
2929 010520 001004                      BNE      74$              ; BR IF NOT
2930 010522 012737 177777 001250 73$:    MOV      #-1,RELEERR     ; SET 'RELEASE ERROR' INDICATOR
2931 010530 104022                      ERROR    22              ; TYPE ERROR MESSAGE 22
2932 010532 013737 001170 001126 74$:    MOV      $TMP2,$BDDAT     ; LOOK FOR BIT FAILURES WHEN RPDS1 READ
2933 010540 013737 001224 001234      MOV      PORTA,PTNBR     ; CHANGE PORT NUMBER
2934 010546 042737 100100 001170      BIC      #ATA!VV,$TMP2    ; DON'T CHECK ATTN BIT OR VV BIT
2935 010554 023737 001124 001170      CMP      $GDDAT,$TMP2    ; ALL BITS OK ?
2936 010562 001401                      BEQ      75$              ; BR IF OK FROM PORT A.
2937 010564 104007                      ERROR    7               ; REPORT ERROR
2938 010566 013737 001172 001126 75$:    MOV      $TMP3,$BDDAT     ; CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2939 010574 013737 001226 001234      MOV      PORTB,PTNBR     ; CHANGE PORT NUMBER
2940 010602 042737 100100 001172      BIC      #ATA!VV,$TMP3    ; DON'T CHECK ATTN BIT OR VV BIT
2941 010610 023737 001124 001172      CMP      $GDDAT,$TMP3    ; SEE IF READ OK FROM PORT B.
2942 010616 001401                      BEQ      76$              ; BR IF OK
2943 010620 104007                      ERROR    7               ; REPORT ERROR
2944 010622 000240 76$:    NOP
2945 010624 000004 3$:    SCOPE                    ; LOOP ?

```

```

*****
*TEST 5          PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'
*****
*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
*FOR THE PORT UNDER TEST.
*
*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
*       DRIVE WAS SEIZED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
*
*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
*       'VV' BIT FOR PORT 'B' WAS SET.
*
*  C.  STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
*       THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
*****

```

```

2963 010626
2964 010626 005737 001274      TST      KYBCTL          ; PERFORMING ONLY SINGLE TESTS ?
2965 010632 001406                      BEQ      2$              ; BR IF NOT
2966 010634 100002                      BPL      1$              ; BR IF JUST ENTERED TEST
2967 010636 000137 002602      JMP      EXEC            ; RETURN & GET NEXT TEST NUMBER
2968 010642 012737 177777 001274 1$:    MOV      #-1,KYBCTL     ; SET SINGLE TEST INDICATOR
2969 010650 112737 000005 001102 2$:    MOVVB   #5,$STNM        ; TEST NUMBER

```

E05

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 56
 CZRJEBO.P11 04-NOV-77 13:27

TS PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

SEQ 0056

```

2970 010656 012737 010700 001106      MOV      #TEST5,$LPADR      ;LOAD LOOP ON TEST ADDRESS
2971 010664 012737 010700 001110      MOV      #TEST5,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
2972 010672 012737 000001 001176      MOV      #1,$TIMES         ;DO 1 ITERATION
2973 010700 012706 001100 000010      MOV      #STACK,SP        ;LOAD THE STACK POINTER
2974 010704 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)    ;SELECT PORT B
2975 010712 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2976
2977      ;*****
2978      ;START THE TIMER
2979
2980 010720 005037 001252      CLR      TIME              ;CLEAR THE ELAPSED TIME COUNTER
2981 010724 012737 003720 001254      MOV      #2000,$WATCH     ;SET WATCH TO 2000 MS
2982 010732 013737 001226 001236      MOV      PORTB,$SEIZPT    ;'SEIZED' PORT ADDRESS
2983
2984      ;*****
2985      ;ISSUE DRIVE CLEAR COMMAND
2986
2987 010740 012760 000011 000000      MOV      #11,RPCS1(RO)    ;ISSUE A DRIVE CLEAR
2988
2989      ;*****
2990      ;VERIFY THAT DRIVE SEIZED BY PORT B.
2991
2992 010746 113760 001224 000010      MOVVB   PORTA,RPCS2(RO)   ;SELECT PORT A
2993 010754 013737 001224 001234      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2994 010762 005037 001244      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
2995 010766 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
2996 010774 012737 000012 001122      MOV      #RPDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2997 011002 060037 001122      ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
2998 011006 005037 001124      CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
2999 011012 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
3000 011020 001403      BEQ      64$              ;BR IF OK
3001 011022 104012      ERROR   12                ;TYPE MESSAGE 12
3002 011024 005137 001244      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
3003 011030 000240      NOP
3004 011032 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)   ;SELECT PORT B
3005 011040 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3006 011046 005037 001244      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
3007 011052 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3008 011060 012737 000012 001122      MOV      #RPDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3009 011066 060037 001122      ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
3010 011072 012737 011600 001124      MOV      #M0L!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
3011 011100 013737 001126 001164      MOV      $BDDAT,$STMP0    ;MOVE REGISTER CONTENTS TO 'STMP0'
3012 011106 042737 106177 001164      BIC     #1C71600,$STMP0   ;SAVE SPECIFIED BITS
3013 011114 023737 001124 001164      CMP     $GDDAT,$STMP0    ;COMPARE THE BITS
3014 011122 001414      BEQ     66$              ;BR IF OK
3015 011124 013737 001126 001174      MOV     $BDDAT,$STMP4     ;COPY 'BAD DATA'
3016 011132 042737 071600 001174      BIC     #71600,$STMP4    ;CLEAR THE MASKED BITS
3017 011140 053737 001174 001124      BIS     $STMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3018 011146 104010      ERROR   10                ;REPORT THE ERROR
3019 011150 005137 001244      COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
3020 011154 000240      NOP
3021 011156 005037 001244      CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
3022 011162 016037 000000 001126      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
3023 011170 012737 000000 001122      MOV     #RPDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3024 011176 060037 001122      ADD     RO,$B0ADR        ;ADD RH11 BASE ADDRESS
3025 011202 012737 004210 001124      MOV     #4210,$GDDAT     ;WHAT REGISTER SHOULD BE
    
```

F05

CZRJEBO, DL CTRLR LGC MACY11 30(1046)
 CZRJEBO.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 57
 T5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

SEQ 0057

3026	011210	013737	001126	001164		MOV	\$BDDAT,\$STMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
3027	011216	042737	100000	001164		BIC	#1C7777,\$STMP0	;SAVE SPECIFIED BITS
3028	011224	023737	001124	001164		CMP	\$GDDAT,\$STMP0	;COMPARE THE BITS
3029	011232	001414				BEQ	68\$;BR IF OK
3030	011234	013737	001126	001174		MOV	\$BDDAT,\$STMP4	;COPY 'BAD DATA'
3031	011242	042737	077777	001174		BIC	#77777,\$STMP4	;CLEAR THE MASKED BITS
3032	011250	053737	001174	001124		BIS	\$STMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3033	011256	104010				ERROR	10	;REPORT THE ERROR
3034	011260	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3035	011264	000240			68\$:	NOP		
3036								
3037								
3038								;;*****
3039								;ISSUE READIN PRESET COMMAND AND SET FMT22
3040	011266	012760	000023	000000		MOV	#23,RPCS1(RO)	;ISSUE A READIN PRESET
3041	011274	012760	010000	000032		MOV	#FMT22,RPOF(RO)	;SET FMT22
3042								
3043								;;*****
3044								;VERIFY THAT THE DRIVE STATUS IS CORRECT
3045								
3046	011302	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3047	011306	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;GET CONTENTS OF RPDS1
3048	011314	012737	000012	001122		MOV	#RPDS1,\$BADDR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3049	011322	060037	001122			ADD	RO,\$BADDR	;ADD RH11 BASE ADDRESS
3050	011326	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;WHAT REGISTER SHOULD BE
3051	011334	013737	001126	001164		MOV	\$BDDAT,\$STMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
3052	011342	042737	106077	001164		BIC	#1C71700,\$STMP0	;SAVE SPECIFIED BITS
3053	011350	023737	001124	001164		CMP	\$GDDAT,\$STMP0	;COMPARE THE BITS
3054	011356	001414				BEQ	70\$;BR IF OK
3055	011360	013737	001126	001174		MOV	\$BDDAT,\$STMP4	;COPY 'BAD DATA'
3056	011366	042737	071700	001174		BIC	#71700,\$STMP4	;CLEAR THE MASKED BITS
3057	011374	053737	001174	001124		BIS	\$STMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3058	011402	104013				ERROR	13	;TYPE MESSAGE 13
3059	011404	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3060	011410	000240			70\$:	NOP		
3061	011412	113760	001224	000010		MOV	PORTA,RPCS2(RO)	;SELECT PORT A
3062	011420	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3063								
3064								;;*****
3065								;WAIT FOR TIMEOUT TO RELEASE DRIVE
3066								
3067	011426	005760	000012		1\$:	TST	RPDS1(RO)	;WAIT FOR THE PORT TO TIME OUT
3068	011432	001006				BNE	2\$;BR WHEN TIMEOUT OCCURS
3069	011434	005737	001254			TST	WATCH	;CHECK THE WATCH
3070	011440	001372				BNE	1\$;BR IF NOT ZERO
3071	011442	104036				ERROR	36	;NO TIMEOUT WITHIN 2 SECONDS
3072	011444	000137	011762			JMP	3\$;BYPASS ATTN REGISTER CHECK
3073								
3074								;;*****
3075								;SEE IF DRIVE RETURNED TO NEUTRAL
3076								
3077	011450				2\$:			
3078								
3079								;VERIFY THAT THE DRIVE IS IN NEUTRAL
3080								
3081	011450	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR

G05

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 58
 CZRJEBO.P11 04-NOV-77 13:27 T5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

SEQ 0058

3082	011454	012737	000012	001122		MOV	#RPDS1,\$BDDADR	FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3083	011462	060037	001122			ADD	RO,\$BDDADR	ADD THE I/O BASE ADDRESS
3084	011466	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY	VV,\$GDDAT ;COMPARISON CONSTANT
3085	011474	113760	001224	000010		MOVB	PORTA,RPDS2(RO)	SELECT PORT A.
3086	011502	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	GET THE DRIVE STATUS REGISTER FROM PORT A.
3087	011510	013737	001170	001164		MOV	\$TMP2,\$TMP0	COPY IT INTO '\$TMP0'
3088	011516	042737	100100	001164		BIC	#ATA!VV,\$TMP0	CLEAR PORT DEPENDENT BITS FROM THE COPY
3089	011524	113760	001226	000010		MOVB	PORTB,RPDS2(RO)	SELECT PORT B.
3090	011532	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	GET THE DRIVE STATUS REGISTER FROM PORT B.
3091	011540	013737	001172	001166		MOV	\$TMP3,\$TMP1	COPY IT INTO '\$TMP1'
3092	011546	042737	100100	001166		BIC	#ATA!VV,\$TMP1	CLEAR PORT DEPENDENT BITS FROM THE COPY
3093	011554	023737	001164	001166		CMP	\$TMP0,\$TMP1	IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3094	011562	001006				BNE	72\$	BR IF NOT
3095	011564	005737	001164			TST	\$TMP0	REGISTERS ARE THE SAME: ARE THEY ZERO ?
3096	011570	001037				BNE	74\$	BR IF NOT
3097	011572	104046				ERROR	46	REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3098	011574	000137	011760			JMP	76\$	BYPASS THE REST OF THE CHECKS
3099	011600	013737	001170	001126	72\$:	MOV	\$TMP2,\$BDDAT	SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3100	011606	013737	001226	001234		MOV	PORTB,PTNBR	SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3101	011614	113760	001226	000010		MOVB	PORTB,RPDS2(RO)	SELECT PORT B.
3102	011622	005737	001164			TST	\$TMP0	SEE IF STATUS EQ 0 FROM PORT A.
3103	011626	001414				BEQ	73\$	BR IF ZERO
3104	011630	013737	001224	001234		MOV	PORTA,PTNBR	SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3105	011636	013737	001172	001126		MOV	\$TMP3,\$BDDAT	'BAD DATA' FOR ERROR TYPE OUT
3106	011644	113760	001224	000010		MOVB	PORTA,RPDS2(RO)	SELECT PORT A.
3107	011652	005737	001166			TST	\$TMP1	SEE IF STATUS EQ ZERO FROM PORT B.
3108	011656	001004				BNE	74\$	BR IF NOT
3109	011660	012737	177777	001250	73\$:	MOV	#-1,RELERR	SET 'RELEASE ERROR' INDICATOR
3110	011666	104026				ERROR	26	TYPE ERROR MESSAGE 26
3111	011670	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT	LOOK FOR BIT FAILURES WHEN RPDS1 READ
3112	011676	013737	001224	001234		MOV	PORTA,PTNBR	CHANGE PORT NUMBER
3113	011704	042737	100000	001170		BIC	#ATA,\$TMP2	DON'T CHECK THE ATTN BIT
3114	011712	023737	001124	001170		CMP	\$GDDAT,\$TMP2	ALL BITS OK ?
3115	011720	001401				BEQ	75\$	BR IF OK FROM PORT A.
3116	011722	104007				ERROR	7	REPORT ERROR
3117	011724	013737	001172	001126	75\$:	MOV	\$TMP3,\$BDDAT	CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3118	011732	013737	001226	001234		MOV	PORTB,PTNBR	CHANGE PORT NUMBER
3119	011740	042737	100000	001172		BIC	#ATA,\$TMP3	DON'T CHECK THE ATTN BIT
3120	011746	023737	001124	001172		CMP	\$GDDAT,\$TMP3	SEE IF READ OK FROM PORT B.
3121	011754	001401				BEQ	76\$	BR IF OK
3122	011756	104007				ERROR	7	REPORT ERROR
3123	011760	000240			76\$:	NOP		
3124	011762	000004			3\$:	SCOPE		;LOOP ?

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

```

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

```

H05

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 59
CZRJEB.P11 04-NOV-77 13:27

T6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

SEG 0059

```

3138
3139 011764
3140 011764 005737 001274
3141 011770 001406
3142 011772 100002
3143 011774 000137 002602
3144 012000 012737 177777 001274
3145 012006 112737 000006 001102
3146 012014 012737 012036 001106
3147 012022 012737 012036 001110
3148 012030 012737 007640 001176
3149 012036 012706 001100
3150
3151
3152
3153
3154 012042 005037 001252
3155 012046 012737 003720 001254
3156
3157
3158
3159
3160
3161 012054 113760 001224 000010
3162 012062 013737 001224 001236
3163 012070 005060 000012
3164 012074 013737 001226 001240
3165
3166
3167
3168
3169
3170 012102 113760 001224 000010
3171 012110 013737 001224 001234
3172 012116 012760 000013 000000
3173
3174
3175
3176 012124 005037 001250
3177 012130 012737 000012 001122
3178 012136 060037 001122
3179 012142 012737 011700 001124
3180 012150 113760 001224 000010
3181 012156 016037 000012 001170
3182 012164 013737 001170 001164
3183 012172 042737 100100 001164
3184 012200 113760 001226 000010
3185 012206 016037 000012 001172
3186 012214 013737 001172 001166
3187 012222 042737 100100 001166
3188 012230 023737 001164 001166
3189 012236 001006
3190 012240 005737 001164
3191 012244 001037
3192 012246 104046
3193 012250 000137 012434

```

```

*****
↑ST6:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #6,$STNM ;TEST NUMBER
MOV #TEST6,$LPAOR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST6,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST6: MOV #STACK,$SP ;LOAD THE STACK POINTER
*****
;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #2000.,WATCH ;SET WATCH TO 2000 MS
*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
*****
;RELEASE THE DRIVE FROM PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A
;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 66$ ;BR IF NOT
TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 68$ ;BR IF NOT
ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
JMP 70$ ;BYPASS THE REST OF THE CHECKS

```

3194	012254	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3195	012262	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3196	012270	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3197	012276	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
3198	012302	001414				BEQ	67\$;BR IF ZERO
3199	012304	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3200	012312	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3201	012320	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3202	012326	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
3203	012332	001004				BNE	68\$;BR IF NOT
3204	012334	012737	177777	001250	67\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
3205	012342	104026				ERROR	26	;TYPE ERROR MESSAGE 26
3206	012344	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3207	012352	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
3208	012360	042737	100000	001170		BIC	#ATA,\$TMP2	;DON'T CHECK THE ATTN BIT
3209	012366	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
3210	012374	001401				BEQ	69\$;BR IF OK FROM PORT A.
3211	012376	104007				ERROR	7	;REPORT ERROR
3212	012400	013737	001172	001126	69\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3213	012406	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
3214	012414	042737	100000	001172		BIC	#ATA,\$TMP3	;DON'T CHECK THE ATTN BIT
3215	012422	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
3216	012430	001401				BEQ	70\$;BR IF OK
3217	012432	104007				ERROR	7	;REPORT ERROR
3218	012434	000240			70\$:	NOP		
3219	012436	005737	001250			TST	RELERR	;DID DRIVE RETURN TO NEUTRAL ?
3220	012442	001402				BEQ	+.6	;BR IF IN NEUTRAL
3221	012444	000137	012720			JMP	1\$;GO WAIT FOR DRIVE TO TIMEOUT
3222	012450	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A
3223	012456	013737	001224	001234		MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3224	012464	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3225	012470	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;GET CONTENTS OF RPDS1
3226	012476	012737	000012	001122		MOV	#RPDS1,\$B0ADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3227	012504	060037	001122			ADD	RO,\$B0ADR	;ADD RH11 BASE ADDRESS
3228	012510	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
3229	012514	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
3230	012522	042737	077777	001164		BIC	#ICATA,\$TMP0	;SAVE SPECIFIED BITS
3231	012530	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
3232	012536	001414				BEQ	71\$;BR IF OK
3233	012540	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
3234	012546	042737	100000	001174		BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
3235	012554	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3236	012562	104017				ERROR	17	;TYPE MESSAGE 17
3237	012564	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3238	012570	000240			71\$:	NOP		
3239	012572	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
3240	012600	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3241	012606	005037	001244			CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
3242	012612	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	;GET CONTENTS OF RPDS1
3243	012620	012737	000012	001122		MOV	#RPDS1,\$B0ADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
3244	012626	060037	001122			ADD	RO,\$B0ADR	;ADD RH11 BASE ADDRESS
3245	012632	005037	001124			CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
3246	012636	013737	001126	001164		MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO 'STMP0'
3247	012644	042737	077777	001164		BIC	#ICATA,\$TMP0	;SAVE SPECIFIED BITS
3248	012652	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
3249	012660	001414				BEQ	73\$;BR IF OK

```

3250 012662 013737 001126 001174      MOV      SBDDAT,$TMP4      ;COPY 'BAD DATA'
3251 012670 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
3252 012676 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3253 012704 104017          ERROR 17                ;TYPE MESSAGE 17
3254 012706 005137 001244          COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3255 012712 000240          NOP
3256 012714 000137 012752          JMP      2$              ;GO CHECK FOR LOOP ON ERROR

```

```

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

```

```

3262 012720          1$:      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
3263 012720 113760 001226 000010      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3264 012726 013737 001226 001234      TST    RPDS1(RO)   ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3265 012734 005760 000012          BNE     2$         ;BR WHEN DRIVE RELEASED
3266 012740 001004          TST    WATCH      ;CHECK THE WATCH
3267 012742 005737 001254          BNE     1$         ;BR IF NOT ZERO
3268 012746 001364          ERROR 36         ;NO TIMEOUT WITHIN 2 SECONDS
3269 012750 104036          SCOPE
3270 012752 000004          ;LOOP ?

```

```

;*****
;TEST 7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

```

```

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

```

```

3284 012754          1$T7:   TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
3285 012754 005737 001274          BEQ     2$         ;BR IF NOT
3286 012760 001406          BPL     1$         ;BR IF JUST ENTERED TEST
3287 012762 100002          JMP     EXEC       ;RETURN & GET NEXT TEST NUMBER
3288 012764 000137 002602          MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3289 012770 012737 177777 001274 1$:      MOVVB   #7,$TSTNM   ;TEST NUMBER
3290 012776 112737 000007 001102 2$:      MOV     #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
3291 013004 012737 013026 001106      MOV     #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3292 013012 012737 013026 001110      MOV     #4000,$TIMES ;DO 4000. ITERATIONS
3293 013020 012737 007640 001176      MOV     #STACK,SP   ;LOAD THE STACK POINTER
3294 013026 012706 001100

```

```

;*****
;START THE TIMER

```

```

3299 013032 005037 001252          CLR     TIME        ;CLEAR THE ELAPSED TIME COUNTER
3300 013036 012737 003720 001254      MOV     #2000.,WATCH ;SET WATCH TO 2000 MS

```

```

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

```

3301
3302
3303
3304
3305

K05

CZRJEBO, DL CTRLR LGC MACY11 30(1046)
 CZRJEBO.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 62
 T7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

SEQ 0062

3306	013044	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	:SELECT PORT B
3307	013052	013737	001226	001236		MOV	PORTB,SEIZPT	:STORE SEIZING PORT'S ADDRESS
3308	013060	005060	000012			CLR	RPDS1(RO)	:WRITE RPDS1
3309	013064	013737	001224	001240		MOV	PORTA,OPRT	: 'OPPOSITE' PORT ADDRESS
3310								
3311								
3312								
3313								
3314								
3315	013072	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	:SELECT PORT B
3316	013100	013737	001226	001234		MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3317	013106	012760	000013	000000		MOV	#13,RPDS1(RO)	:ISSUE RELEASE THROUGH PORT B
3318								
3319								
3320								
3321	013114	005037	001250			CLR	RELERR	:CLEAR THE 'RELEASE ERROR' INDICATOR
3322	013120	012737	000012	001122		MOV	#RPDS1,\$BDADR	:FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3323	013126	060037	001122			ADD	RO,\$BDADR	:ADD THE I/O BASE ADDRESS
3324	013132	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	:COMPARISON CONSTANT
3325	013140	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	:SELECT PORT A.
3326	013146	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	:GET THE DRIVE STATUS REGISTER FROM PORT A.
3327	013154	013737	001170	001164		MOV	\$TMP2,\$TMP0	:COPY IT INTO '\$TMP0'
3328	013162	042737	100100	001164		BIC	#ATA!VV,\$TMP0	:CLEAR PORT DEPENDENT BITS FROM THE COPY
3329	013170	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	:SELECT PORT B.
3330	013176	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	:GET THE DRIVE STATUS REGISTER FROM PORT B.
3331	013204	013737	001172	001166		MOV	\$TMP3,\$TMP1	:COPY IT INTO '\$TMP1'
3332	013212	042737	100100	001166		BIC	#ATA!VV,\$TMP1	:CLEAR PORT DEPENDENT BITS FROM THE COPY
3333	013220	023737	001164	001166		CMP	\$TMP0,\$TMP1	:IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3334	013226	001006				BNE	66\$:BR IF NOT
3335	013230	005737	001164			TST	\$TMP0	:REGISTERS ARE THE SAME: ARE THEY ZERO ?
3336	013234	001037				BNE	68\$:BR IF NOT
3337	013236	104046				ERROR	46	:REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3338	013240	000137	013424			JMP	70\$:BYPASS THE REST OF THE CHECKS
3339	013244	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDAT	:SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3340	013252	013737	001226	001234		MOV	PORTB,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3341	013260	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	:SELECT PORT B.
3342	013266	005737	001164			TST	\$TMP0	:SEE IF STATUS EQ 0 FROM PORT A.
3343	013272	001414				BEQ	67\$:BR IF ZERO
3344	013274	013737	001224	001234		MOV	PORTA,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3345	013302	013737	001172	001126		MOV	\$TMP3,\$BDAT	: 'BAD DATA' FOR ERROR TYPE OUT
3346	013310	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	:SELECT PORT A.
3347	013316	005737	001166			TST	\$TMP1	:SEE IF STATUS EQ ZERO FROM PORT B.
3348	013322	001004				BNE	68\$:BR IF NOT
3349	013324	012737	177777	001250	67\$:	MOV	#-1,RELERR	:SET 'RELEASE ERROR' INDICATOR
3350	013332	104026				ERROR	26	:TYPE ERROR MESSAGE 26
3351	013334	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDAT	:LOOK FOR BIT FAILURES WHEN RPDS1 READ
3352	013342	013737	001224	001234		MOV	PORTA,PTNBR	:CHANGE PORT NUMBER
3353	013350	042737	100000	001170		BIC	#ATA,\$TMP2	:DON'T CHECK THE ATTN BIT
3354	013356	023737	001124	001170		CMP	\$GDDAT,\$TMP2	:ALL BITS OK ?
3355	013364	001401				BEQ	69\$:BR IF OK FROM PORT A.
3356	013366	104007				ERROR	7	:REPORT ERROR
3357	013370	013737	001172	001126	69\$:	MOV	\$TMP3,\$BDAT	:CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3358	013376	013737	001226	001234		MOV	PORTB,PTNBR	:CHANGE PORT NUMBER
3359	013404	042737	100000	001172		BIC	#ATA,\$TMP3	:DON'T CHECK THE ATTN BIT
3360	013412	023737	001124	001172		CMP	\$GDDAT,\$TMP3	:SEE IF READ OK FROM PORT B.
3361	013420	001401				BEQ	70\$:BR IF OK

L05

CZRJEBO DL CTRLR LGC MACY11 30(1046)
 CZRJEB.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 63
 T7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

SEQ 0063

```

3362 013422 104007
3363 013424 000240
3364 013426 005737 001250
3365 013432 001402
3366 013434 000137 013710
3367 013440 113760 001226 000010
3368 013446 013737 001226 001234
3369 013454 005037 001244
3370 013460 016037 000012 001126
3371 013466 012737 000012 001122
3372 013474 060037 001122
3373 013500 005037 001124
3374 013504 013737 001126 001164
3375 013512 042737 077777 001164
3376 013520 023737 001124 001164
3377 013526 001414
3378 013530 013737 001126 001174
3379 013536 042737 100000 001174
3380 013544 053737 001174 001124
3381 013552 104017
3382 013554 005137 001244
3383 013560 000240
3384 013562 113760 001224 000010
3385 013570 013737 001224 001234
3386 013576 005037 001244
3387 013602 016037 000012 001126
3388 013610 012737 000012 001122
3389 013616 060037 001122
3390 013622 005037 001124
3391 013626 013737 001126 001164
3392 013634 042737 077777 001164
3393 013642 023737 001124 001164
3394 013650 001414
3395 013652 013737 001126 001174
3396 013660 042737 100000 001174
3397 013666 053737 001174 001124
3398 013674 104017
3399 013676 005137 001244
3400 013702 000240
3401 013704 000137 013742
3402
3403
3404
3405
3406
3407 013710
3408 013710 113760 001224 000010
3409 013716 013737 001224 001234
3410 013724 005760 000012
3411 013730 001004
3412 013732 005737 001254
3413 013736 001364
3414 013740 104036
3415 013742 000004
3416
3417

70$: ERROR 7 ;REPORT ERROR
NOP
TST RELEARR ;DID DRIVE RETURN TO NEUTRAL ?
BEQ +6 ;BR IF IN NEUTRAL
JMP 1$ ;GO WAIT FOR DRIVE TO TIMEOUT
MOV PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
CLR SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,$TMP0 ;COMPARE THE BITS
BEQ 71$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #ATA,$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

71$: NOP
MOV PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
CLR SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,$TMP0 ;COMPARE THE BITS
BEQ 73$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #ATA,$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

73$: NOP
JMP 2$ ;GO CHECK FOR LOOP ON ERROR

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

1$: MOV PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RPDS1(RO) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
BNE 2$ ;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 ?

```

MOS

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 64
CZRJEB.P11 04-NOV-77 13:27

T10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL

SEQ 0064

3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473

013744
013744 005737 001274
013750 001406
013752 100002
013754 000137 002602
013760 012737 177777 001274
013766 112737 000010 001102
013774 012737 014016 001106
014002 012737 014016 001110
014010 012737 000144 001176
014016 012706 001100
014022 113760 001224 000010
014030 013737 001224 001234
014036 013737 001224 001236

014044 012760 000013 000000

014052 005037 001250
014056 012737 000012 001122
014064 060037 001122
014070 012737 011700 001124
014076 113760 001224 000010
014104 016037 000012 001170
014112 013737 001170 001164
014120 042737 100100 001164
014126 113760 001226 000010
014134 016037 000012 001172
014142 013737 001172 001166
014150 042737 100100 001166
014156 023737 001164 001166
014164 001006
014166 005737 001164
014172 001037
014174 104046
014176 000137 014362
014202 013737 001170 001126
014210 013737 001226 001234
014216 113760 001226 000010

*TEST 10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
*
*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
*
* A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
* NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
*

TST10:
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 #10,\$STSTNM ;TEST NUMBER
MOV #TEST10,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST10,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #100,\$TIMES ;DO 100. ITERATIONS
TEST10: MOV #STACK,SP ;LOAD THE STACK POINTER
MOVB PORTA,RPDS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

;ISSUE A RELEASE COMMAND

MOV #13,RPDS1(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 #RPDS1,\$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,\$BDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL:PGM:DPR:DRY:VV,\$GDDAT ;COMPARISON CONSTANT
MOVB PORTA,RPDS2(RO) ;SELECT PORT A.
MOV RPDS1(RO),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV \$TMP2,\$TMP0 ;COPY IT INTO 'TMP0'
BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,RPDS2(RO) ;SELECT PORT B.
MOV RPDS1(RO),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV \$TMP3,\$TMP1 ;COPY IT INTO 'TMP1'
BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 64\$;BR IF NOT
TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 66\$;BR IF NOT
ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
JMP 68\$;BYPASS THE REST OF THE CHECKS
64\$: MOV \$TMP2,\$BDADR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOVB PORTB,RPDS2(RO) ;SELECT PORT B.

N05

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 65
CZRJEB.P11 04-NOV-77 13:27 T10

TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL

SEQ 0065

```

3474 014224 005737 001164      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
3475 014230 001414      BEQ      65$        ;BR IF ZERO
3476 014232 013737 001224 001234  MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3477 014240 013737 001172 001126  MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3478 014246 113760 001224 000010  MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
3479 014254 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
3480 014260 001004      BNE      66$        ;BR IF NOT
3481 014262 012737 177777 001250 65$:    MOV      #-1,RELEARR ;SET 'RELEASE ERROR' INDICATOR
3482 014270 104030      ERROR   30          ;TYPE ERROR MESSAGE 30
3483 014272 013737 001170 001126 66$:    MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3484 014300 013737 001224 001234  MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
3485 014306 042737 100000 001170  BIC      #ATA,$TMP2  ;DON'T CHECK THE ATTN BIT
3486 014314 023737 001124 001170  CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
3487 014322 001401      BEQ      67$        ;BR IF OK FROM PORT A.
3488 014324 104007      ERROR   7          ;REPORT ERROR
3489 014326 013737 001172 001126 67$:    MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3490 014334 013737 001226 001234  MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
3491 014342 042737 100000 001172  BIC      #ATA,$TMP3  ;DON'T CHECK THE ATTN BIT
3492 014350 023737 001124 001172  CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3493 014356 001401      BEQ      68$        ;BR IF OK
3494 014360 104007      ERROR   7          ;REPORT ERROR
3495 014362 000240 68$:    NOP
3496 014364 000004      SCOPE

```

;LOOP ?

;TEST 11 TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL

;TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

;ST11:

```

3507 014366      TST      KYBCTL     ;PERFORMING ONLY SINGLE TESTS ?
3508 014366 005737 001274      BEQ      2$        ;BR IF NOT
3509 014372 001406      BPL      1$        ;BR IF JUST ENTERED TEST
3510 014374 100002      JMP      EXEC      ;RETURN & GET NEXT TEST NUMBER
3511 014376 000137 002602      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3512 014402 012737 177777 001274 1$:    MOV      #11,$STNM  ;TEST NUMBER
3513 014410 112737 000011 001102 2$:    MOVVB   #TEST11,$LPADR ;LOAD LOOP ON TEST ADDRESS
3514 014416 012737 014440 001106  MOV      #TEST11,$LPERR ;LOAD LOOP ON ERROR ADDRESS
3515 014424 012737 014440 001110  MOV      #100,$TIMES ;DO 100. ITERATIONS
3516 014432 012737 000144 001176  TEST11: MOV      #STACK,SP ;LOAD THE STACK POINTER
3517 014440 012706 001100      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
3518 014444 113760 001226 000010  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3519 014452 013737 001226 001234  MOV      PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3520 014460 013737 001226 001236

```

;ISSUE A RELEASE COMMAND

```

3524 014466 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

3521
3522
3523
3524
3525
3526
3527
3528
3529

```

3530 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3531 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
3532 MOV #RPDS1,SBADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3533 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
3534 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
3535 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
3536 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3537 STMP2,STMP0 ;COPY IT INTO 'STMP0'
3538 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3539 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
3540 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3541 STMP3,STMP1 ;COPY IT INTO 'STMP1'
3542 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3543 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3544 BNE 64$ ;BR IF NOT
3545 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3546 BNE 66$ ;BR IF NOT
3547 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3548 JMP 68$ ;BYPASS THE REST OF THE CHECKS
3549 MOV STMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3550 MOVB PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3551 PORTB,RPCS2(RO) ;SELECT PORT B.
3552 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3553 BEQ 65$ ;BR IF ZERO
3554 MOVB PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3555 STMP3,SBDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3556 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
3557 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3558 BNE 66$ ;BR IF NOT
3559 MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3560 ERROR 30 ;TYPE ERROR MESSAGE 30
3561 MOV STMP2,SBDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3562 PORTA,PTNBR ;CHANGE PORT NUMBER
3563 BIC #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
3564 CMP $GDDAT,STMP2 ;ALL BITS OK ?
3565 BEQ 67$ ;BR IF OK FROM PORT A.
3566 ERROR 7 ;REPORT ERROR
3567 MOV STMP3,SBDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3568 PORTB,PTNBR ;CHANGE PORT NUMBER
3569 BIC #ATA,STMP3 ;DON'T CHECK THE ATTN BIT
3570 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
3571 BEQ 68$ ;BR IF OK
3572 ERROR 7 ;REPORT ERROR
3573 NOP
3574 SCOPE
3575 ;LOOP ?

```

```

3577
3578
3579
3580 *****
3581 *TEST 12 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
3582 *
3583 *VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3584 * PORT TO RELEASE THE DRIVE.
3585 *
3586 * A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'A'.

```



```

3642 ;*****
3643 ;DRIVE CLEAR THROUGH PORT A FIRST
3644
3645 015264 012760 000011 000000          MOV    #11,RPCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT A
3646
3647 ;*****
3648 ;VERIFY THAT DRIVE STILL SEIZED BY PORT A
3649
3650 015272 113760 001226 000010          MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
3651 015300 013737 001226 001234          MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3652 015306 005037 001244                    CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3653 015312 016037 000012 001126          MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
3654 015320 012737 000012 001122          MOV    #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3655 015326 060037 001122                    ADD    RO,$BDADR ;ADD RH11 BASE ADDRESS
3656 015332 005037 001124                    CLR    $GDDAT ;WHAT REGISTER SHOULD BE
3657 015336 013737 001126 001164          MOV    $BDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
3658 015344 042737 100000 001164          BIC    #1C7777,$TMPD ;SAVE SPECIFIED BITS
3659 015352 023737 001124 001164          CMP    $GDDAT,$TMPD ;COMPARE THE BITS
3660 015360 001414                    BEQ    66$ ;BR IF OK
3661 015362 013737 001126 001174          MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
3662 015370 042737 077777 001174          BIC    #77777,$TMP4 ;CLEAR THE MASKED BITS
3663 015376 053737 001174 001124          BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3664 015404 104033                    ERROR  33 ;TYPE MESSAGE 33
3665 015406 005137 001244                    COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3666 015412 000240          66$: NOP
3667 015414 113760 001224 000010          MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
3668 015422 013737 001224 001234          MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3669 015430 005037 001244                    CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3670 015434 016037 000012 001126          MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
3671 015442 012737 000012 001122          MOV    #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3672 015450 060037 001122                    ADD    RO,$BDADR ;ADD RH11 BASE ADDRESS
3673 015454 012737 011700 001124          MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
3674 015462 013737 001126 001164          MOV    $BDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
3675 015470 042737 100000 001164          BIC    #1C7777,$TMPD ;SAVE SPECIFIED BITS
3676 015476 023737 001124 001164          CMP    $GDDAT,$TMPD ;COMPARE THE BITS
3677 015504 001414                    BEQ    68$ ;BR IF OK
3678 015506 013737 001126 001174          MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
3679 015514 042737 077777 001174          BIC    #77777,$TMP4 ;CLEAR THE MASKED BITS
3680 015522 053737 001174 001124          BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3681 015530 104033                    ERROR  33 ;TYPE MESSAGE 33
3682 015532 005137 001244                    COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3683 015536 000240          68$: NOP
3684
3685 ;*****
3686 ;NOW ISSUE MASSBUS INIT
3687
3688 015540 012760 000040 000010          MOV    #CLR,RPCS2(RO) ;ISSUE MASSBUS INIT
3689
3690 ;*****
3691 ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
3692
3693 015546 113760 001226 000010          MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
3694 015554 013737 001226 001234          MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3695 015562 005037 001244                    CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3696 015566 016037 000012 001126          MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
3697 015574 012737 000012 001122          MOV    #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

3698	015602	060037	001122		ADD	RO, \$BDADR	; ADD RH11 BASE ADDRESS
3699	015606	005037	001124		CLR	\$GDDAT	; WHAT REGISTER SHOULD BE
3700	015612	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'TMP0'
3701	015620	042737	100000	001164	BIC	#1C7777, \$TMP0	; SAVE SPECIFIED BITS
3702	015626	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
3703	015634	001414			BEQ	70\$; BR IF OK
3704	015636	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
3705	015644	042737	077777	001174	BIC	#77777, \$TMP4	; CLEAR THE MASKED BITS
3706	015652	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3707	015660	104034			ERROR	34	; TYPE MESSAGE 34
3708	015662	005137	001244		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
3709	015666	000240			70\$:	NOP	
3710	015670	113760	001224	000010	MOVB	PORTA, \$RPCS2(RO)	; SELECT PORT A
3711	015676	013737	001224	001234	MOV	PORTA, \$PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3712	015704	005037	001244		CLR	CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
3713	015710	016037	000012	001126	MOV	\$RPS1(RO), \$BDDAT	; GET CONTENTS OF \$RPS1
3714	015716	012737	000012	001122	MOV	\$RPS1, \$BDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
3715	015724	060037	001122		ADD	RO, \$BDADR	; ADD RH11 BASE ADDRESS
3716	015730	012737	011700	001124	MOV	\$MOL!PGM!DPR!DRY!VV, \$GDDAT	; WHAT REGISTER SHOULD BE
3717	015736	013737	001126	001164	MOV	\$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO 'TMP0'
3718	015744	042737	100000	001164	BIC	#1C7777, \$TMP0	; SAVE SPECIFIED BITS
3719	015752	023737	001124	001164	CMP	\$GDDAT, \$TMP0	; COMPARE THE BITS
3720	015760	001414			BEQ	72\$; BR IF OK
3721	015762	013737	001126	001174	MOV	\$BDDAT, \$TMP4	; COPY 'BAD DATA'
3722	015770	042737	077777	001174	BIC	#77777, \$TMP4	; CLEAR THE MASKED BITS
3723	015776	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3724	016004	104034			ERROR	34	; TYPE MESSAGE 34
3725	016006	005137	001244		COM	CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
3726	016012	000240			72\$:	NOP	
3727							
3728							; RELEASE THE DRIVE FROM PORT A
3729							
3730	016014	113760	001224	000010	MOVB	PORTA, \$RPCS2(RO)	; SELECT PORT A
3731	016022	013737	001224	001234	MOV	PORTA, \$PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3732	016030	012760	000013	000000	MOV	#13, \$RPS1(RO)	; ISSUE RELEASE THROUGH PORT A
3733							
3734							; VERIFY THAT THE DRIVE IS IN NEUTRAL
3735							
3736	016036	005037	001250		CLR	RELERR	; CLEAR THE 'RELEASE ERROR' INDICATOR
3737	016042	012737	000012	001122	MOV	\$RPS1, \$BDADR	; FORM THE ADDRESS OF \$RPS1 FOR TYPEOUT
3738	016050	060037	001122		ADD	RO, \$BDADR	; ADD THE I/O BASE ADDRESS
3739	016054	012737	011700	001124	MOV	\$MOL!PGM!DPR!DRY!VV, \$GDDAT	; COMPARISON CONSTANT
3740	016062	113760	001224	000010	MOVB	PORTA, \$RPCS2(RO)	; SELECT PORT A.
3741	016070	016037	000012	001170	MOV	\$RPS1(RO), \$TMP2	; GET THE DRIVE STATUS REGISTER FROM PORT A.
3742	016076	013737	001170	001164	MOV	\$TMP2, \$TMP0	; COPY IT INTO 'TMP0'
3743	016104	042737	100100	001164	BIC	\$ATA!VV, \$TMP0	; CLEAR PORT DEPENDENT BITS FROM THE COPY
3744	016112	113760	001226	000010	MOVB	PORTB, \$RPCS2(RO)	; SELECT PORT B.
3745	016120	016037	000012	001172	MOV	\$RPS1(RO), \$TMP3	; GET THE DRIVE STATUS REGISTER FROM PORT B.
3746	016126	013737	001172	001166	MOV	\$TMP3, \$TMP1	; COPY IT INTO 'TMP1'
3747	016134	042737	100100	001166	BIC	\$ATA!VV, \$TMP1	; CLEAR PORT DEPENDENT BITS FROM THE COPY
3748	016142	023737	001164	001166	CMP	\$TMP0, \$TMP1	; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3749	016150	001006			BNE	74\$; BR IF NOT
3750	016152	005737	001164		TST	\$TMP0	; REGISTERS ARE THE SAME: ARE THEY ZERO ?
3751	016156	001045			BNE	76\$; BR IF NOT
3752	016160	104046			ERROR	46	; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3753	016162	000137	016362		JMP	78\$; BYPASS THE REST OF THE CHECKS

```

3754 016166 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3755 016174 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3756 016202 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
3757 016210 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
3758 016214 001414 BEQ 75$ ;BR IF ZERO
3759 016216 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3760 016224 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3761 016232 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
3762 016240 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
3763 016244 001012 BNE 76$ ;BR IF NOT
3764 016246 012737 177777 001250 75$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
3765 016254 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
3766 016262 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
3767 016270 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
3768 016272 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3769 016300 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
3770 016306 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
3771 016314 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
3772 016322 001401 BEQ 77$ ;BR IF OK FROM PORT A.
3773 016324 104007 ERROR 7 ;REPORT ERROR
3774 016326 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3775 016334 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
3776 016342 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
3777 016350 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3778 016356 001401 BEQ 78$ ;BR IF OK
3779 016360 104007 ERROR 7 ;REPORT ERROR
3780 016362 000240 78$: NOP
3781 016364 000004 1$: SCOPE ;LOOP ?

```

```

3782
3783 *****
3784 *TEST 13 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
3785 *
3786 *VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3787 * PORT TO RELEASE THE DRIVE.
3788 *
3789 * A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.
3790 * VERIFY THAT THE DRIVE HAS BEEN SEIZED.
3791 *
3792 * B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
3793 * DOES NOT RETURN TO NEUTRAL.
3794 *
3795 * C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
3796 * DOES NOT RETURN TO NEUTRAL.
3797 *
3798 * D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
3799 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3800 *
3801 *****

```

```

3802 016366 005737 001274 †ST13: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
3803 016366 001406 BEQ 2$ ;BR IF NOT
3804 016372 100002 BPL 1$ ;BR IF JUST ENTERED TEST
3805 016374 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
3806 016402 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
3807 016410 112737 000013 001102 2$: MOVB #13,$STNM ;TEST NUMBER
3808 016416 012737 016440 001106 MOV #TEST13,$LPADR ;LOAD LOOP ON TEST ADDRESS

```



```

3810 016424 012737 016440 001110
3811 016432 012737 007640 001176
3812 016440 012706 001100
3813
3814
3815
3816
3817
3818 016444 113760 001226 000010
3819 016452 013737 001226 001236
3820 016460 005060 000012
3821 016464 113760 001224 000010
3822 016472 013737 001224 001234
3823 016500 013737 001224 001240
3824 016506 016037 000012 001126
3825 016514 010037 001122
3826 016520 062737 000012 001122
3827 016526 005037 001124
3828 016532 023737 001124 001126
3829 016540 001403
3830 016542 104004
3831 016544 000137 017742
3832 016550
3833 016550 113760 001226 000010
3834 016556 013737 001226 001234
3835 016564 016037 000012 001126
3836 016572 012737 011700 001124
3837 016600 013737 001124 001166
3838 016606 005137 001166
3839 016612 013737 001126 001164
3840 016620 043737 001166 001164
3841 016626 023737 001124 001164
3842 016634 001401
3843 016636 104005
3844 016640 000240
3845
3846
3847
3848
3849 016642 012760 000011 000000
3850
3851
3852
3853
3854 016650 113760 001224 000010
3855 016656 013737 001224 001234
3856 016664 005037 001244
3857 016670 016037 000012 001126
3858 016676 012737 000012 001122
3859 016704 060037 001122
3860 016710 005037 001124
3861 016714 013737 001126 001164
3862 016722 042737 100000 001164
3863 016730 023737 001124 001164
3864 016736 001414
3865 016740 013737 001126 001174

```

```

MOV #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST13: MOV #STACK,SP ;LOAD THE STACK POINTER
;*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$BDADR ;RH11 BASE ADDRESS
ADD #RPDS1,$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 1$ ;BYPASS REST OF THE SUBTEST
64$:
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!V!$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$:
NOP
;*****
;DRIVE CLEAR THROUGH PORT B FIRST
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT B
;*****
;VERIFY THAT DRIVE STILL SEIZED BY PORT B
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$BDADR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'

```

```

3866 016746 042737 077777 001174 BIC #77777,$TMP4 ;CLEAR THE MASKED BITS
3867 016754 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3868 016762 104033 ERROR 33 ;TYPE MESSAGE 33
3869 016764 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3870 016770 000240 66$: NOP
3871 016772 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
3872 017000 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3873 017006 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3874 017012 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
3875 017020 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3876 017026 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
3877 017032 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
3878 017040 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3879 017046 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
3880 017054 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
3881 017062 001414 BEQ 68$ ;BR IF OK
3882 017064 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
3883 017072 042737 077777 001174 BIC #77777,$TMP4 ;CLEAR THE MASKED BITS
3884 017100 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3885 017106 104033 ERROR 33 ;TYPE MESSAGE 33
3886 017110 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3887 017114 000240 68$: NOP
3888
3889 ;:*****
3890 ;NOW ISSUE MASSBUS INIT
3891
3892 017116 012760 000040 000010 MOV #CLR,RPCS2(RO) ;ISSUE MASSBUS INIT
3893
3894 ;:*****
3895 ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
3896
3897 017124 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
3898 017132 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3899 017140 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3900 017144 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
3901 017152 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3902 017160 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
3903 017164 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
3904 017170 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3905 017176 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
3906 017204 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
3907 017212 001414 BEQ 70$ ;BR IF OK
3908 017214 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
3909 017222 042737 077777 001174 BIC #77777,$TMP4 ;CLEAR THE MASKED BITS
3910 017230 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3911 017236 104034 ERROR 34 ;TYPE MESSAGE 34
3912 017240 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3913 017244 000240 70$: NOP
3914 017246 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
3915 017254 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3916 017262 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3917 017266 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
3918 017274 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3919 017302 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
3920 017306 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
3921 017314 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'

```

3922	017322	042737	100000	001164		BIC	#1C77777,\$TMP0	;SAVE SPECIFIED BITS
3923	017330	023737	001124	001164		CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
3924	017336	001414				BEQ	72\$;BR IF OK
3925	017340	013737	001126	001174		MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
3926	017346	042737	077777	001174		BIC	#77777,\$TMP4	;CLEAR THE MASKED BITS
3927	017354	053737	001174	001124		BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
3928	017362	104034				ERROR	34	;TYPE MESSAGE 34
3929	017364	005137	001244			COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
3930	017370	000240			72\$:	NOP		
3931								
3932								;RELEASE THE DRIVE FROM PORT B
3933								
3934	017372	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B
3935	017400	013737	001226	001234		MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3936	017406	012760	000013	000000		MOV	#13,RPCS1(RO)	;ISSUE RELEASE THROUGH PORT B
3937								
3938								;VERIFY THAT THE DRIVE IS IN NEUTRAL
3939								
3940	017414	005037	001250			CLR	RELERR	;CLEAR THE 'RELEASE ERROR' INDICATOR
3941	017420	012737	000012	001122		MOV	#RPDS1,\$BDDADR	;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3942	017426	060037	001122			ADD	RO,\$BDDADR	;ADD THE I/O BASE ADDRESS
3943	017432	012737	011700	001124		MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
3944	017440	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3945	017446	016037	000012	001170		MOV	RPDS1(RO),\$TMP2	;GET THE DRIVE STATUS REGISTER FROM PORT A.
3946	017454	013737	001170	001164		MOV	\$TMP2,\$TMP0	;COPY IT INTO '\$TMP0'
3947	017462	042737	100100	001164		BIC	#ATA!VV,\$TMP0	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3948	017470	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3949	017476	016037	000012	001172		MOV	RPDS1(RO),\$TMP3	;GET THE DRIVE STATUS REGISTER FROM PORT B.
3950	017504	013737	001172	001166		MOV	\$TMP3,\$TMP1	;COPY IT INTO '\$TMP1'
3951	017512	042737	100100	001166		BIC	#ATA!VV,\$TMP1	;CLEAR PORT DEPENDENT BITS FROM THE COPY
3952	017520	023737	001164	001166		CMP	\$TMP0,\$TMP1	;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3953	017526	001006				BNE	74\$;BR IF NOT
3954	017530	005737	001164			TST	\$TMP0	;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3955	017534	001045				BNE	76\$;BR IF NOT
3956	017536	104046				ERROR	46	;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3957	017540	000137	017740			JMP	78\$;BYPASS THE REST OF THE CHECKS
3958	017544	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3959	017552	013737	001226	001234		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3960	017560	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT B.
3961	017566	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
3962	017572	001414				BEQ	75\$;BR IF ZERO
3963	017574	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3964	017602	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
3965	017610	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
3966	017616	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
3967	017622	001012				BNE	76\$;BR IF NOT
3968	017624	012737	177777	001250	75\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
3969	017632	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
3970	017640	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
3971	017646	104026				ERROR	26	;TYPE ERROR MESSAGE 26
3972	017650	013737	001170	001126	76\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3973	017656	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
3974	017664	042737	100000	001170		BIC	#ATA,\$TMP2	;DON'T CHECK THE ATTN BIT
3975	017672	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
3976	017700	001401				BEQ	77\$;BR IF OK FROM PORT A.
3977	017702	104007				ERROR	7	;REPORT ERROR

```

3978 017704 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3979 017712 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
3980 017720 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
3981 017726 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
3982 017734 001401 BEQ 78$ ;BR IF OK
3983 017736 104007 ERROR 7 ;REPORT ERROR
3984 017740 000240 78$: NOP
3985 017742 000004 1$: SCOPE ;LOOP ?

```

3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033

```

*****
*TEST 14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
*
*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
* SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* C. ISSUE A MASSBUS CLEAR.
*
* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION
* BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
* 'B' IS STILL SET.
*****

```

```

*****
†ST14:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #14,$STSTN ;TEST NUMBER
MOV #TEST14,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST14,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST14: MOV #STACK,$SP ;LOAD THE STACK POINTER
;*****

```

;SET ATTENTION BITS FOR BOTH PORTS

```

MOVB PORTA,RPCS2(RO) ;SELECT PORT 64$
MOV #-1,RPER1(RO) ;FORCE ERRORS
CLR RPER1(RO) ;CLEAR THE ERRORS
MOV PORTB,RPCS2(RO) ;SELECT THE OTHER PORT
64$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RPER1(RO) ;FORCE ERRORS ON PORT 65$
CLR RPER1(RO) ;CLEAR THE ERRORS
MOVB PORTA,RPCS2(RO) ;SELECT PORT "64$" AGAIN
65$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT

```

```

4034
4035
4036
4037 020104 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
4038 020112 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4039 020120 005037 001244          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4040 020124 016037 000012 001126      MOV    RPDS1(RO),SBDAT ;GET CONTENTS OF RPDS1
4041 020132 012737 000012 001122      MOV    #RPDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4042 020140 060037 001122          ADD    RO,SBDADR ;ADD RH11 BASE ADDRESS
4043 020144 012737 100000 001124      MOV    #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
4044 020152 013737 001126 001164      MOV    SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4045 020160 042737 077777 001164      BIC    #CATA,$TMP0 ;SAVE SPECIFIED BITS
4046 020166 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
4047 020174 001414          BEQ    66$ ;BR IF OK
4048 020176 013737 001126 001174      MOV    SBDAT,$TMP4 ;COPY 'BAD DATA'
4049 020204 042737 100000 001174      BIC    #ATA,$TMP4 ;CLEAR THE MASKED BITS
4050 020212 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4051 020220 104010          ERROR 10 ;REPORT THE ERROR
4052 020222 005137 001244          COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4053 020226 000240          NOP
4054 020230 005737 001244          TST    CKERR ;WAS ATTN BIT FOR PORT A SET ?
4055 020234 001402          BEQ    .+6 ;BR IF IT WAS
4056 020236 000137 021244          JMP    1$ ;BYPASS REST OF TEST IF NOT
4057 020242 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
4058 020250 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4059 020256 005037 001244          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4060 020262 016037 000012 001126      MOV    RPDS1(RO),SBDAT ;GET CONTENTS OF RPDS1
4061 020270 012737 000012 001122      MOV    #RPDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4062 020276 060037 001122          ADD    RO,SBDADR ;ADD RH11 BASE ADDRESS
4063 020302 012737 100000 001124      MOV    #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
4064 020310 013737 001126 001164      MOV    SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4065 020316 042737 077777 001164      BIC    #CATA,$TMP0 ;SAVE SPECIFIED BITS
4066 020324 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
4067 020332 001414          BEQ    68$ ;BR IF OK
4068 020334 013737 001126 001174      MOV    SBDAT,$TMP4 ;COPY 'BAD DATA'
4069 020342 042737 100000 001174      BIC    #ATA,$TMP4 ;CLEAR THE MASKED BITS
4070 020350 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4071 020356 104010          ERROR 10 ;REPORT THE ERROR
4072 020360 005137 001244          COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4073 020364 000240          NOP
4074 020366 005737 001244          TST    CKERR ;WAS ATTN BIT FOR PORT B SET ?
4075 020372 001402          BEQ    .+6 ;BR IF IT WAS
4076 020374 000137 021244          JMP    1$ ;BYPASS REST OF TEST IF NOT
4077
4078
4079
4080
4081 ;SEIZE THE DRIVE THROUGH PORT A
4082 020400 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
4083 020406 013737 001224 001236      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
4084 020414 005060 000012          CLR    RPDS1(RO) ;WRITE RPDS1
4085 020420 013737 001226 001240      MOV    PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
4086
4087
4088 ;ISSUE MASSBUS INIT TO PORT A
4089

```

```

4090 020426 012760 000040 000010      MOV      #CLR,RPCS2(R0) ;MASSBUS INIT
4091 020434 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A AGAIN
4092
4093
4094
4095
4096 020442 005037 001244      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
4097 020446 016037 000012 001126      MOV      RPDS1(R0),SDDAT ;GET CONTENTS OF RPDS1
4098 020454 012737 000012 001122      MOV      #RPDS1,SBDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4099 020462 060037 001122      ADD     RO,SBDADR      ;ADD RH11 BASE ADDRESS
4100 020466 005037 001124      CLR      SGGDAT        ;WHAT REGISTER SHOULD BE
4101 020472 013737 001126 001164      MOV      SDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
4102 020500 042737 077777 001164      BIC     #1CATA,$TMP0  ;SAVE SPECIFIED BITS
4103 020506 023737 001124 001164      CMP     SGGDAT,$TMP0  ;COMPARE THE BITS
4104 020514 001414      BEQ     72$           ;BR IF OK
4105 020516 013737 001126 001174      MOV      SDDAT,$TMP4   ;COPY 'BAD DATA'
4106 020524 042737 100000 001174      BIC     #ATA,$TMP4    ;CLEAR THE MASKED BITS
4107 020532 053737 001174 001124      BIS     $TMP4,$GGDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
4108 020540 104047      ERROR  47           ;TYPE MESSAGE 47
4109 020542 005137 001244      COM     CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
4110 020546 000240      NOP
4111
4112
4113
4114
4115
4116 020550 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A
4117 020556 013737 001224 001234      MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4118 020564 012760 000013 000000      MOV      #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A
4119
4120
4121
4122 020572 005037 001250      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
4123 020576 012737 000012 001122      MOV      #RPDS1,SBDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4124 020604 060037 001122      ADD     RO,SBDADR      ;ADD THE I/O BASE ADDRESS
4125 020610 012737 011700 001124      MOV      #MOL:PGM:DPR:DRY:VV,$GGDAT ;COMPARISON CONSTANT
4126 020616 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
4127 020624 016037 000012 001170      MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4128 020632 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
4129 020640 042737 100100 001164      BIC     #ATA:VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4130 020646 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
4131 020654 016037 000012 001172      MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4132 020662 013737 001172 001166      MOV      $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
4133 020670 042737 100100 001166      BIC     #ATA:VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4134 020676 023737 001164 001166      CMP     $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4135 020704 001006      BNE     74$          ;BR IF NOT
4136 020706 005737 001164      TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4137 020712 001045      BNE     76$          ;BR IF NOT
4138 020714 104046      ERROR  46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4139 020716 000137 021116      JMP     78$          ;BYPASS THE REST OF THE CHECKS
4140 020722 013737 001170 001126 74$: MOV      $TMP2,SDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4141 020730 013737 001226 001234      MOV      PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4142 020736 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
4143 020744 005737 001164      TST     $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
4144 020750 001414      BEQ     75$          ;BR IF ZERO
4145 020752 013737 001224 001234      MOV      PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    
```

```

4146 020760 013737 001172 001126      MOV      $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
4147 020766 113760 001224 000010      MOVB     PORTA,RPCS2(RO)   ;SELECT PORT A.
4148 020774 005737 001166      TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
4149 021000 001012 177777 001250 75$:      BNE      76$              ;BR IF NOT
4150 021002 012737 000011 000000      MOV      #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
4151 021010 012760 000011 000000      MOV      #11,RPCS1(RO)    ;CLEAR THE DRIVE
4152 021016 012760 000013 000000      MOV      #13,RPCS1(RO)    ;RELEASE THE DRIVE
4153 021024 104026 001170 001126 76$:      ERROR    26              ;TYPE ERROR MESSAGE 26
4154 021026 013737 001224 001234      MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4155 021034 013737 100000 001170      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
4156 021042 042737 001124 001170      BIC      #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
4157 021050 023737 001124 001170      CMP      $GDDAT,$TMP2     ;ALL BITS OK ?
4158 021056 001401 001172 001126 77$:      BEQ      77$              ;BR IF OK FROM PORT A.
4159 021060 104007 001172 001126      ERROR    7               ;REPORT ERROR
4160 021062 013737 001226 001234      MOV      $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4161 021070 013737 100000 001172      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
4162 021076 042737 001124 001172      BIC      #ATA,$TMP3       ;DON'T CHECK THE ATTN BIT
4163 021104 023737 001124 001172      CMP      $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
4164 021112 001401 001172 001172      BEQ      78$              ;BR IF OK
4165 021114 104007 001172 001172      ERROR    7               ;REPORT ERROR
4166 021116 000240 001172 001172 78$:      NOP
4167
4168
4169
4170
4171 021120 113760 001226 000010      MOVB     PORTB,RPCS2(RO)   ;SELECT PORT B
4172 021126 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4173 021134 005037 001244 001122      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
4174 021140 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
4175 021146 012737 000012 001122      MOV      #RPDS1,$BDAOR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4176 021154 060037 001122 001124      ADD      RO,$BDAOR        ;ADD RH11 BASE ADDRESS
4177 021160 012737 100000 001124      MOV      #ATA,$GDDAT      ;WHAT REGISTER SHOULD BE
4178 021166 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
4179 021174 042737 077777 001164      BIC      #1CAT,$TMP0      ;SAVE SPECIFIED BITS
4180 021202 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
4181 021210 001414 001126 001174      BEQ      79$              ;BR IF OK
4182 021212 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
4183 021220 042737 100000 001174      BIC      #ATA,$TMP4       ;CLEAR THE MASKED BITS
4184 021226 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
4185 021234 104050 001244 001124      ERROR    50              ;TYPE MESSAGE 50
4186 021236 005137 001244 001124      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
4187 021242 000240 001244 001124 79$:      NOP
4188 021244 000004 001244 001124 1$:      SCOPE                    ;LOOP ?
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201

```

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

;TEST 15 TEST RESET ATTENTION 'B' BY MASSBUS CLEAR

- *****
- ;VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.
- *****
- ;A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- *****
- ;B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- *****
- ;C. ISSUE A MASSBUS CLEAR.
- *****

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

†ST15:

```
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 0-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB 15,$STNM ;TEST NUMBER
MOV 0TEST15,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV 0TEST15,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV 04,$TIMES ;DO 4 ITERATIONS
TEST15: MOV 0STACK,$SP ;LOAD THE STACK POINTER
;*****
```

;SET ATTENTION BITS FOR BOTH PORTS

```
MOV B PORTA,RPCS2(RO) ;SELECT PORT 64$
MOV 0-1,RPER1(RO) ;FORCE ERRORS
CLR RPER1(RO) ;CLEAR THE ERRORS
MOV PORTB,RPCS2(RO) ;SELECT THE OTHER PORT
64$: TST RPS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
MOV 0-1,RPER1(RO) ;FORCE ERRORS ON PORT 65$
CLR RPER1(RO) ;CLEAR THE ERRORS
65$: MOV B PORTA,RPCS2(RO) ;SELECT PORT "64$" AGAIN
TST RPS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```
MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPS1(RO),$BDDAT ;GET CONTENTS OF RPS1
MOV 0RPS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$B0ADR ;ADD RHI1 BASE ADDRESS
MOV 0ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC 0ICATA,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
BIC 0ATA,$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
66$: NOP
TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
BEQ +6 ;BR IF IT WAS
JMP 15 ;BYPASS REST OF TEST IF NOT
```

4202
4203
4204
4205
4206
4207
4208 021246 005737 001274
4209 021246 001406
4210 021252 100002
4211 021254 000137 002602
4212 021256 012737 177777 001274
4213 021262 012737 000015 001102
4214 021270 112737 021320 001106
4215 021276 012737 021320 001110
4216 021304 012737 000004 001176
4217 021312 012706 001100
4218 021320
4219
4220
4221
4222
4223 021324 113760 001224 000010
4224 021332 012760 177777 000014
4225 021340 005060 000014
4226 021344 013760 001226 000010
4227 021352 005760 000012
4228 021356 001775
4229 021360 012760 177777 000014
4230 021366 005060 000014
4231 021372 113760 001224 000010
4232 021400 005760 000012
4233 021404 001775
4234
4235
4236
4237
4238 021406 113760 001226 000010
4239 021414 013737 001226 001234
4240 021422 005037 001244
4241 021426 016037 000012 001126
4242 021434 012737 000012 001122
4243 021442 060037 001122
4244 021446 012737 100000 001124
4245 021454 013737 001126 001164
4246 021462 042737 077777 001164
4247 021470 023737 001124 001164
4248 021476 001414
4249 021500 013737 001126 001174
4250 021506 042737 100000 001174
4251 021514 053737 001174 001124
4252 021522 104010
4253 021524 005137 001244
4254 021530 000240
4255 021532 005737 001244
4256 021536 001402
4257 021540 000137 022546


```

4258 021544 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
4259 021552 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4260 021560 005037 001244      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4261 021564 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
4262 021572 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4263 021600 060037 001122      ADD    RO,SBADR ;ADD RHI1 BASE ADDRESS
4264 021604 012737 100000 001124      MOV    #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
4265 021612 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4266 021620 042737 077777 001164      BIC    #CATA,$TMP0 ;SAVE SPECIFIED BITS
4267 021626 023737 001124 001164      CMP    SGDDAT,$TMP0 ;COMPARE THE BITS
4268 021634 001414      BEQ    68$ ;BR IF OK
4269 021636 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
4270 021644 042737 100000 001174      BIC    #ATA,$TMP4 ;CLEAR THE MASKED BITS
4271 021652 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4272 021660 104010      ERROR  10 ;REPORT THE ERROR
4273 021662 005137 001244      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4274 021666 000240      68$:  NOP
4275 021670 005737 001244      TST    CKERR ;WAS ATTN BIT FOR PORT A SET ?
4276 021674 001402      BEQ    +6 ;BR IF IT WAS
4277 021676 000137 022546      JMP    1$ ;BYPASS REST OF TEST IF NOT
4278
4279 ;*****
4280
4281 ;SEIZE THE DRIVE THROUGH PORT B
4282
4283 021702 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
4284 021710 013737 001226 001236      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
4285 021716 005060 000012      CLR    RPDS1(RO) ;WRITE RPDS1
4286 021722 013737 001224 001240      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
4287
4288 ;*****
4289 ;ISSUE MASSBUS INIT TO PORT B
4290
4291 021730 012760 000040 000010      MOV    #CLR,RPCS2(RO) ;MASSBUS INIT
4292 021736 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B AGAIN
4293
4294 ;*****
4295 ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
4296
4297 021744 005037 001244      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4298 021750 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
4299 021756 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4300 021764 060037 001122      ADD    RO,SBADR ;ADD RHI1 BASE ADDRESS
4301 021770 005037 001124      CLR    SGDDAT ;WHAT REGISTER SHOULD BE
4302 021774 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
4303 022002 042737 077777 001164      BIC    #CATA,$TMP0 ;SAVE SPECIFIED BITS
4304 022010 023737 001124 001164      CMP    SGDDAT,$TMP0 ;COMPARE THE BITS
4305 022016 001414      BEQ    72$ ;BR IF OK
4306 022020 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
4307 022026 042737 100000 001174      BIC    #ATA,$TMP4 ;CLEAR THE MASKED BITS
4308 022034 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4309 022042 104047      ERROR  47 ;TYPE MESSAGE 47
4310 022044 005137 001244      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4311 022050 000240      72$:  NOP
4312
4313 ;*****

```

```

4314 ;RELEASE THE DRIVE FROM PORT B
4315
4316
4317 022052 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
4318 022060 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4319 022066 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4320
4321 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4322
4323 022074 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4324 022100 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4325 022106 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
4326 022112 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4327 022120 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
4328 022126 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4329 022134 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
4330 022142 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4331 022150 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
4332 022156 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4333 022164 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
4334 022172 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4335 022200 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4336 022206 001006 BNE 74$ ;BR IF NOT
4337 022210 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4338 022214 001045 BNE 76$ ;BR IF NOT
4339 022216 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4340 022220 000137 022420 JMP 78$ ;BYPASS THE REST OF THE CHECKS
4341 022224 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4342 022232 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4343 022240 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
4344 022246 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4345 022252 001414 BEQ 75$ ;BR IF ZERO
4346 022254 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4347 022262 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4348 022270 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
4349 022276 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4350 022302 001012 BNE 76$ ;BR IF NOT
4351 022304 012737 177777 001250 75$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4352 022312 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
4353 022320 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
4354 022326 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
4355 022330 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4356 022336 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
4357 022344 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
4358 022352 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
4359 022360 001401 BEQ 77$ ;BR IF OK FROM PORT A.
4360 022362 104007 ERROR 7 ;REPORT ERROR
4361 022364 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4362 022372 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4363 022400 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
4364 022406 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
4365 022414 001401 BEQ 78$ ;BR IF OK
4366 022416 104007 ERROR 7 ;REPORT ERROR
4367 022420 000240 78$: NOP
4368
4369

```

::*****

```

4370
4371
4372 022422 113760 001224 000010
4373 022430 013737 001224 001234
4374 022436 005037 001244
4375 022442 016037 000012 001126
4376 022450 012737 000012 001122
4377 022456 060037 001122
4378 022462 012737 100000 001124
4379 022470 013737 001126 001164
4380 022476 042737 077777 001164
4381 022504 023737 001124 001164
4382 022512 001414
4383 022514 013737 001126 001174
4384 022522 042737 100000 001174
4385 022530 053737 001174 001124
4386 022536 104050
4387 022540 005137 001244
4388 022544 000240
4389 022546 000004
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407 022550
4408 022550 005737 001274
4409 022554 001406
4410 022556 100002
4411 022560 000137 002602
4412 022564 012737 177777 001274
4413 022572 112737 000016 001102
4414 022600 012737 022622 001106
4415 022606 012737 022622 001110
4416 022614 012737 000004 001176
4417 022622 012706 001100
4418
4419
4420
4421
4422 022626 113760 001224 000010
4423 022634 012760 177777 000014
4424 022642 005060 000014
4425 022646 013760 001226 000010

```

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

```

MOV B PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(R0),SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,SBDAOR ;ADD RHI1 BASE ADDRESS
MOV #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,$TMP0 ;COMPARE THE BITS
BEQ 79$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 50 ;TYPE MESSAGE 50
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
SCOPE ;LOOP ?
79$:
1$:

```

```

*****
*TEST 16 TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
*
*VERIFY THAT MASSBUS CLEAR DOES NOT RESET ATTENTION BITS WHEN THE
* DRIVE IS IN NEUTRAL.
*
* A. SET THE ATTENTION BITS FOR BOTH PORTS.
*
* B. VERIFY THAT THE DRIVE IS IN NEUTRAL.
*
* C. ISSUE A MASSBUS INIT. VERIFY THAT NEITHER ATTENTION BIT HAS
* RESET.
*****

```

```

*****
†ST16:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #16,$STNM ;TEST NUMBER
MOV #TEST16,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST16,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST16: MOV #STACK,$SP ;LOAD THE STACK POINTER
*****

```

;SET ATTENTION BITS FOR BOTH PORTS

```

MOV B PORTA,RPCS2(R0) ;SELECT PORT 64$
MOV #-1,RPER1(R0) ;FORCE ERRORS
CLR RPER1(R0) ;CLEAR THE ERRORS
MOV PORTB,RPCS2(R0) ;SELECT THE OTHER PORT

```

```

4426 022654 005760 000012 64$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
4427 022660 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
4428 022662 012760 177777 000014 MOV #-1, RPER1(RO) ;FORCE ERRORS ON PORT 65$
4429 022670 005060 000014 CLR RPER1(RO) ;CLEAR THE ERRORS
4430 022674 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT "64$" AGAIN
4431 022702 005760 000012 65$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
4432 022706 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
4433
4434 ;:*****
4435 ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
4436
4437 022710 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT A
4438 022716 013737 001224 001234 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4439 022724 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4440 022730 016037 000012 001126 MOV RPDS1(RO), SBDDAT ;GET CONTENTS OF RPDS1
4441 022736 012737 000012 001122 MOV #RPDS1, SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4442 022744 060037 001122 ADD RO, SBADR ;ADD RH11 BASE ADDRESS
4443 022750 012737 100000 001124 MOV #ATA, SGDDAT ;WHAT REGISTER SHOULD BE
4444 022756 013737 001126 001164 MOV SBDDAT, STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4445 022764 042737 077777 001164 BIC #ICATA, STMP0 ;SAVE SPECIFIED BITS
4446 022772 023737 001124 001164 CMP SGDDAT, STMP0 ;COMPARE THE BITS
4447 023000 001414 BEQ 66$ ;BR IF OK
4448 023002 013737 001126 001174 MOV SBDDAT, STMP4 ;COPY 'BAD DATA'
4449 023010 042737 100000 001174 BIC #ATA, STMP4 ;CLEAR THE MASKED BITS
4450 023016 053737 001174 001124 BIS STMP4, SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4451 023024 104010 ERROR 10 ;REPORT THE ERROR
4452 023026 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4453 023032 000240 66$: NOP
4454 023034 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
4455 023040 001402 BEQ +6 ;BR IF IT WAS
4456 023042 000137 024006 JMP IS ;BYPASS REST OF TEST IF NOT
4457 023046 113760 001226 000010 MOVB PORTB, RPCS2(RO) ;SELECT PORT B
4458 023054 013737 001226 001234 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4459 023062 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
4460 023066 016037 000012 001126 MOV RPDS1(RO), SBDDAT ;GET CONTENTS OF RPDS1
4461 023074 012737 000012 001122 MOV #RPDS1, SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4462 023102 060037 001122 ADD RO, SBADR ;ADD RH11 BASE ADDRESS
4463 023106 012737 100000 001124 MOV #ATA, SGDDAT ;WHAT REGISTER SHOULD BE
4464 023114 013737 001126 001164 MOV SBDDAT, STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
4465 023122 042737 077777 001164 BIC #ICATA, STMP0 ;SAVE SPECIFIED BITS
4466 023130 023737 001124 001164 CMP SGDDAT, STMP0 ;COMPARE THE BITS
4467 023136 001414 BEQ 68$ ;BR IF OK
4468 023140 013737 001126 001174 MOV SBDDAT, STMP4 ;COPY 'BAD DATA'
4469 023146 042737 100000 001174 BIC #ATA, STMP4 ;CLEAR THE MASKED BITS
4470 023154 053737 001174 001124 BIS STMP4, SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
4471 023162 104010 ERROR 10 ;REPORT THE ERROR
4472 023164 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
4473 023170 000240 68$: NOP
4474 023172 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
4475 023176 001402 BEQ +6 ;BR IF IT WAS
4476 023200 000137 024006 JMP IS ;BYPASS REST OF TEST IF NOT
4477
4478 ;:*****
4479 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4480
4481

```

```

4482 023204 005037 001250          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
4483 023210 012737 000012 001122    MOV      #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4484 023216 060037 001122          ADD      R0,$BDADR      ;ADD THE I/O BASE ADDRESS
4485 023222 012737 111700 001124    MOV      #111700,$GDDAT ;COMPARISON CONSTANT
4486 023230 113760 001224 000010    MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
4487 023236 016037 000012 001170    MOV      RPDS1(R0),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4488 023244 013737 001170 001164    MOV      STMP2,STMP0    ;COPY IT INTO 'STMP0'
4489 023252 042737 100100 001164    BIC     #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4490 023260 113760 001226 000010    MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
4491 023266 016037 000012 001172    MOV      RPDS1(R0),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4492 023274 013737 001172 001166    MOV      STMP3,STMP1    ;COPY IT INTO 'STMP1'
4493 023302 042737 100100 001166    BIC     #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4494 023310 023737 001164 001166    CMP     STMP0,STMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4495 023316 001006          BNE     70$            ;BR IF NOT
4496 023320 005737 001164          TST     STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4497 023324 001045          BNE     72$            ;BR IF NOT
4498 023326 104046          ERROR   46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4499 023330 000137 023514          JMP     74$            ;BYPASS THE REST OF THE CHECKS
4500 023334 013737 001170 001126 70$:   MOV      STMP2,$BDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4501 023342 013737 001226 001234    MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4502 023350 113760 001226 000010    MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
4503 023356 005737 001164          TST     STMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
4504 023362 001414          BEQ     71$            ;BR IF ZERO
4505 023364 013737 001224 001234    MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4506 023372 013737 001172 001126    MOV      STMP3,$BDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4507 023400 113760 001224 000010    MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
4508 023406 005737 001166          TST     STMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
4509 023412 001012          BNE     72$            ;BR IF NOT
4510 023414 012737 177777 001250 71$:   MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
4511 023422 012760 000011 000000    MOV      #11,RPCS1(R0)  ;CLEAR THE DRIVE
4512 023430 012760 000013 000000    MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
4513 023436 104026          ERROR   26            ;TYPE ERROR MESSAGE 26
4514 023440 013737 001170 001126 72$:   MOV      STMP2,$BDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4515 023446 013737 001224 001234    MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
4516 023454 023737 001124 001170    CMP     $GDDAT,STMP2   ;ALL BITS OK ?
4517 023462 001401          BEQ     73$            ;BR IF OK FROM PORT A.
4518 023464 104007          ERROR   7            ;REPORT ERROR
4519 023466 013737 001172 001126 73$:   MOV      STMP3,$BDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4520 023474 013737 001226 001234    MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
4521 023502 023737 001124 001172    CMP     $GDDAT,STMP3   ;SEE IF READ OK FROM PORT B.
4522 023510 001401          BEQ     74$            ;BR IF OK
4523 023512 104007          ERROR   7            ;REPORT ERROR
4524 023514 000240 74$:   NOP
4525 023516 005737 001250          TST     RELERR          ;WAS DRIVE IN NEUTRAL ?
4526 023522 001402          BEQ     +6            ;BR IF IT WAS
4527 023524 000137 024006          JMP     1$            ;BYPASS RESET OF TEST
4528
4529 ;*****
4530 ;ISSUE THE MASSBUS INIT
4531 023530 012760 000040 000010          MOV      #CLR,RPCS2(R0) ;ISSUE A MASSBUS INIT
4532
4533 ;*****
4534 ;CHECK THE ATTENTION BITS OF BOTH PORTS
4535
4536 023536 113760 001224 000010    MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
4537 023544 013737 001224 001234    MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

4538 023552 005037 001244
4539 023556 016037 000012 001126
4540 023564 012737 000012 001122
4541 023572 060037 001122
4542 023576 012737 100000 001124
4543 023604 013737 001126 001164
4544 023612 042737 077777 001164
4545 023620 023737 001124 001164
4546 023626 001414
4547 023630 013737 001126 001174
4548 023636 042737 100000 001174
4549 023644 053737 001174 001124
4550 023652 104051
4551 023654 005137 001244
4552 023660 000240
4553 023662 113760 001226 000010
4554 023670 013737 001226 001234
4555 023676 005037 001244
4556 023702 016037 000012 001126
4557 023710 012737 000012 001122
4558 023716 060037 001122
4559 023722 012737 100000 001124
4560 023730 013737 001126 001164
4561 023736 042737 077777 001164
4562 023744 023737 001124 001164
4563 023752 001414
4564 023754 013737 001126 001174
4565 023762 042737 100000 001174
4566 023770 053737 001174 001124
4567 023776 104051
4568 024000 005137 001244
4569 024004 000240
4570 024006 000004
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585 024010
4586 024010 005737 001274
4587 024014 001406
4588 024016 100002
4589 024020 000137 002602
4590 024024 012737 177777 001274
4591 024032 112737 000017 001102
4592 024040 012737 024062 001106
4593 024046 012737 024062 001110

```

```

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO) SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
MOV #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CATA,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 75$ ;BR IF OK
MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 51 ;TYPE MESSAGE 51
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
75$: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO) SBDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
MOV #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CATA,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 77$ ;BR IF OK
MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 51 ;TYPE MESSAGE 51
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
77$:
1$: SCOPE ;LOOP ?

```

```

*****
*TEST 17 TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'
*
*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
*
* A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'A'; VERIFY THAT
* THE DRIVE IS SEIZED.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****

```

```

↑ST17: 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 #17,STSTNM ;TEST NUMBER
MOV #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS

```

```

4594 024054 012737 007640 001176
4595 024062 012706 001100
4596
4597
4598
4599 024066 113760 001224 000010
4600 024074 005060 000012
4601 024100 012760 000011 000000
4602 024106 012760 000013 000000
4603 024114 113760 001226 000010
4604 024122 005060 000012
4605 024126 012760 000011 000000
4606 024134 012760 000013 000000
4607
4608
4609
4610
4611
4612 024142 113760 001224 000010
4613 024150 013737 001224 001236
4614 024156 005760 000000
4615 024162 113760 001226 000010
4616 024170 013737 001226 001234
4617 024176 013737 001226 001240
4618 024204 016037 000012 001126
4619 024212 010037 001122
4620 024216 062737 000012 001122
4621 024224 005037 001124
4622 024230 023737 001124 001126
4623 024236 001403
4624 024240 104004
4625 024242 000137 024674
4626 024246
4627 024246 113760 001224 000010
4628 024254 013737 001224 001234
4629 024262 016037 000012 001126
4630 024270 012737 011700 001124
4631 024276 013737 001124 001166
4632 024304 005137 001166
4633 024310 013737 001126 001164
4634 024316 043737 001166 001164
4635 024324 023737 001124 001164
4636 024332 001401
4637 024334 104005
4638 024336 000240
4639
4640
4641
4642
4643
4644 024340 113760 001224 000010
4645 024346 013737 001224 001234
4646 024354 012760 000013 000000
4647
4648
4649

```

```

MOV #4000, $TIMES ; DO 4000. ITERATIONS
TEST17: MOV #STACK, SP ; LOAD THE STACK POINTER

; CLEAR ATTENTION BITS FOR BOTH PORTS

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

; *****

; SEIZE THE DRIVE THROUGH PORT A

MOVB PORTA, RPCS2(RO) ; SELECT PORT A
MOV PORTA, SEIZPT ; STORE SEIZING PORT'S ADDRESS
RPCS1(RO) ; READ RHCS1
MOVB PORTB, RPCS2(RO) ; SELECT PORT B
MOV PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB, OPPRT ; 'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO), $BDDAT ; SEE IF DRIVE SEIZED BY PORT A
MOV RO, $BDADR ; R#11 BASE ADDRESS
ADD #RPDS1, $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 1$ ; BYPASS REST OF THE SUBTEST

64$:
MOVB PORTA, RPCS2(RO) ; SELECT PORT A
MOV PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO), $BDDAT ; SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ; EXPECTED STATUS
MOV $GDDAT, $TMP1 ; USE GOOD DATA AS A MASK
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

; *****

; RELEASE THE DRIVE FROM PORT A

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

; VERIFY THAT THE DRIVE IS IN NEUTRAL

```

```

4650 024362 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4651 024366 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4652 024374 060037 001122 ADD RD,$BDDADR ;ADD THE I/O BASE ADDRESS
4653 024400 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
4654 024406 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
4655 024414 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4656 024422 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
4657 024430 042737 100100 001164 BIC #ATA:VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4658 024436 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
4659 024444 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4660 024452 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
4661 024460 042737 100100 001166 BIC #ATA:VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4662 024466 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4663 024474 001006 BNE 66$ ;BR IF NOT
4664 024476 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4665 024502 001045 BNE 68$ ;BR IF NOT
4666 024504 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4667 024506 000137 024672 JMP 70$ ;BYPASS THE REST OF THE CHECKS
4668 024512 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4669 024520 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4670 024526 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
4671 024534 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4672 024540 001414 BEQ 67$ ;BR IF ZERO
4673 024542 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4674 024550 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4675 024556 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
4676 024564 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4677 024570 001012 BNE 68$ ;BR IF NOT
4678 024572 012737 177777 001250 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4679 024600 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
4680 024606 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
4681 024614 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
4682 024616 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4683 024624 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
4684 024632 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
4685 024640 001401 BEQ 69$ ;BR IF OK FROM PORT A.
4686 024642 104007 ERROR 7 ;REPORT ERROR
4687 024644 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4688 024652 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
4689 024660 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
4690 024666 001401 BEQ 70$ ;BR IF OK
4691 024670 104007 ERROR 7 ;REPORT ERROR
4692 024672 000240 70$: NOP
4693 024674 000004 1$: SCOPE ;LOOP ?

```

```

4694
4695 ;*****
4696 ;*TEST 20 TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
4697 ;*
4698 ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
4699 ;*
4700 ;* A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'B'; VERIFY THAT
4701 ;* THE DRIVE IS SEIZED.
4702 ;*
4703 ;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
4704 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4705 ;*

```



```

4706
4707 024676
4708 024676 005737 001274
4709 024702 001406
4710 024704 100002
4711 024706 000137 002602
4712 024712 012737 177777 001274 1$:
4713 024720 112737 000020 001102 2$:
4714 024726 012737 024750 001106
4715 024734 012737 024750 001110
4716 024742 012737 007640 001176
4717 024750 012706 001100
4718
4719
4720
4721 024754 113760 001224 000010
4722 024762 005060 000012
4723 024766 012760 000011 000000
4724 024774 012760 000013 000000
4725 025002 113760 001226 000010
4726 025010 005060 000012
4727 025014 012760 000011 000000
4728 025022 012760 000013 000000
4729
4730
4731
4732
4733
4734 025030 113760 001226 000010
4735 025036 013737 001226 001236
4736 025044 005760 000000
4737 025050 113760 001224 000010
4738 025056 013737 001224 001234
4739 025064 013737 001224 001240
4740 025072 016037 000012 001126
4741 025100 010037 001122
4742 025104 062737 000012 001122
4743 025112 005037 001124
4744 025116 023737 001124 001126
4745 025124 001403
4746 025126 104004
4747 025130 000137 025562
4748 025134
4749 025134 113760 001226 000010 64$:
4750 025142 013737 001226 001234
4751 025150 016037 000012 001126
4752 025156 012737 011700 001124
4753 025164 013737 001124 001166
4754 025172 005137 001166
4755 025176 013737 001126 001164
4756 025204 043737 001166 001164
4757 025212 023737 001124 001164
4758 025220 001401
4759 025222 104005
4760 025224 000240 65$:
4761

```

```

*****
↑ST20:
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 #20,$STNM ;TEST NUMBER
MOV #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST20: MOV #STACK,$P ;LOAD THE STACK POINTER

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

*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RPCS1(RO) ;READ RHCS1
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$BDADR ;RH11 BASE ADDRESS
ADD #RPDS1,$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 1$ ;BYPASS REST OF THE SUBTEST

64$: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
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

```

;;*****

```

4762
4763
4764
4765
4766 025226 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
4767 025234 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4768 025242 012760 000013 000000      MOV    #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
4769
4770
4771
4772 025250 005037 001250          CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
4773 025254 012737 000012 001122      MOV    #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4774 025262 060037 001122          ADD    RO,$BDDADR ;ADD THE I/O BASE ADDRESS
4775 025266 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
4776 025274 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
4777 025302 016037 000012 001170      MOV    RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4778 025310 013737 001170 001164      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
4779 025316 042737 100100 001164      BIC    #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4780 025324 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
4781 025332 016037 000012 001172      MOV    RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4782 025340 013737 001172 001166      MOV    STMP3,STMP1 ;COPY IT INTO 'STMP1'
4783 025346 042737 100100 001166      BIC    #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4784 025354 023737 001164 001166      CMP    STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4785 025362 001006          BNE    66$ ;BR IF NOT
4786 025364 005737 001164          TST    STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4787 025370 001045          BNE    68$ ;BR IF NOT
4788 025372 104046          ERROR  46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4789 025374 000137 025560          JMP    70$ ;BYPASS THE REST OF THE CHECKS
4790 025400 013737 001170 001126 66$:      MOV    STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4791 025406 013737 001226 001234      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4792 025414 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
4793 025422 005737 001164          TST    STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
4794 025426 001414          BEQ    67$ ;BR IF ZERO
4795 025430 013737 001224 001234      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4796 025436 013737 001172 001126      MOV    STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
4797 025444 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
4798 025452 005737 001166          TST    STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
4799 025456 001012          BNE    68$ ;BR IF NOT
4800 025460 012737 177777 001250 67$:      MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
4801 025466 012760 000011 000000      MOV    #11,RPCS1(RO) ;CLEAR THE DRIVE
4802 025474 012760 000013 000000      MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
4803 025502 104026          ERROR  26 ;TYPE ERROR MESSAGE 26
4804 025504 013737 001170 001126 68$:      MOV    STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4805 025512 013737 001224 001234      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
4806 025520 023737 001124 001170      CMP    $GDDAT,STMP2 ;ALL BITS OK ?
4807 025526 001401          BEQ    69$ ;BR IF OK FROM PORT A.
4808 025530 104007          ERROR  7 ;REPORT ERROR
4809 025532 013737 001172 001126 69$:      MOV    STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4810 025540 013737 001226 001234      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
4811 025546 023737 001124 001172      CMP    $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
4812 025554 001401          BEQ    70$ ;BR IF OK
4813 025556 104007          ERROR  7 ;REPORT ERROR
4814 025560 000240          NOP
4815 025562 000004          IS:   SCOPE
4816
4817
;LOOP ?

```

4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873

025564
025564 005737 001274
025570 001406
025572 100002
025574 000137 002602
025600 012737 177777 001274
025606 112737 000021 001102
025614 012737 025636 001106
025622 012737 025636 001110
025630 012737 007640 001176
025636 012706 001100

025642 113760 001224 000010
025650 005060 000012
025654 012760 000011 000000
025662 012760 000013 000000
025670 113760 001226 000010
025676 005060 000012
025702 012760 000011 000000
025710 012760 000013 000000

025716 113760 001226 000010
025724 013737 001226 001236
025732 005060 000012
025736 013737 001224 001240
025744 113760 001224 000010
025752 013737 001224 001234

```
*****  
*TEST 21 TEST 'PORT REQUEST' FROM PORT 'A'  
*  
*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE  
* DRIVE IS SEIZED BY THE OTHER PORT.  
*  
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.  
*  
* B. WRITE 0'S INTO RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL  
* SEIZED BY PORT 'B'.  
*  
* C. ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE  
* SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR  
* PORT 'A' AND IS NOT SET FOR PORT 'B'.  
*  
* D. ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE  
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.  
*  
*****  
TST21:  
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 #21,$TSTNM ;TEST NUMBER  
MOV #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS  
MOV #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS  
MOV #4000,$TIMES ;DO 4000. ITERATIONS  
TEST21: MOV #STACK,SP ;LOAD THE STACK POINTER  
  
;CLEAR ATTENTION BITS FOR BOTH PORTS  
MOVB PORTA,RPCS2(R0) ;SELECT PORT #A  
CLR RPDS1(R0) ;SEIZE THE DRIVE  
MOV #11,RPCS1(R0) ;ISSUE DRIVE CLEAR  
MOV #13,RPCS1(R0) ;RELEASE THE DRIVE  
MOVB PORTB,RPCS2(R0) ;SELECT PORT #B  
CLR RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'  
MOV #11,RPCS1(R0) ;ISSUE DRIVE CLEAR  
MOV #13,RPCS1(R0) ;RELEASE THE DRIVE  
  
;*****  
;SEIZE THE DRIVE THROUGH PORT B  
MOVB PORTB,RPCS2(R0) ;SELECT PORT B  
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS  
CLR RPDS1(R0) ;WRITE RPDS1  
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS  
MOVB PORTA,RPCS2(R0) ;SELECT PORT A  
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
  
;*****  
;SET PORT REQUEST
```

4874 025760 005060 000012

CLR RPDS1(RO) ;SET PORT REQUEST FOR PORT A

4875
4876
4877
4878
4879

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

;RELEASE THE DRIVE FROM PORT B

4880
4881
4882 025764 113760 001226 000010
4883 025772 013737 001226 001234
4884 026000 012760 000013 000000

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT B

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

4885
4886
4887
4888 026006 005037 001250
4889 026012 012737 111700 001124
4890 026020 012737 000012 001122
4891 026026 060037 001122
4892 026032 113760 001224 000010
4893 026040 013737 001224 001234
4894 026046 016037 000012 001164
4895 026054 113760 001226 000010
4896 026062 013737 001226 001234
4897 026070 016037 000012 001126
4898 026076 001404
4899 026100 005737 001154
4900 026104 001401
4901 026106 104031
4902 026110 013737 001164 001126
4903 026116 013737 001224 001234
4904 026124 023737 001124 001126
4905 026132 001401
4906 026134 104027
4907 026136 000240

CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
MOV #ATA!MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
MOV #RPDS1,\$BADDR ;REGISTER ADDRESS INCREMENT
ADD RO,\$BADDR ;REGISTER BASE ADDRESS FOR TYPEOUT
MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),STMP0 ;READ STATUS REGISTER FROM PORT A
MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(RO),\$BDDAT ;DRIVE STATUS FROM PORT B
BEQ 66\$;BR IF STATUS FROM PORT B ZERO
TST STMP0 ;IS STATUS FROM PORT A ZERO ?
BEQ 66\$;BR IF ZERO
ERROR 31 ;REPORT DRIVE IN NEUTRAL
MOV STMP0,\$BDDAT ;CHECK STATUS FROM PORT A
MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
CMP SGDDAT,\$BDDAT ;COMPARE WITH CONSTANT
BEQ 67\$;BR IF OK
ERROR 27 ;REPORT REGISTER ERROR

66\$:

67\$:

4908 026140 113760 001226 000010
4909 026146 013737 001226 001234
4910 026154 005037 001244
4911 026160 016037 000012 001126
4912 026166 012737 000012 001122
4913 026174 060037 001122
4914 026200 005037 001124
4915 026204 013737 001126 001164
4916 026212 042737 077777 001164
4917 026220 023737 001124 001164
4918 026226 001414
4919 026230 013737 001126 001174
4920 026236 042737 100000 001174
4921 026244 053737 001174 001124
4922 026252 104016
4923 026254 005137 001244

MOV B PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),\$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,\$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,\$BADDR ;ADD RH11 BASE ADDRESS
CLR SGDDAT ;WHAT REGISTER SHOULD BE
MOV \$BDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CATA,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 68\$;BR IF OK
MOV \$BDDAT,STMP4 ;COPY 'BAD DATA'
BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 16 ;TYPE MESSAGE 16
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

68\$:

4924 026260 000240
4925 026262 113760 001224 000010
4926 026270 013737 001224 001234
4927 026276 005037 001244
4928 026302 016037 000012 001126
4929 026310 012737 000012 001122

NOP
MOV B PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO),\$BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1,\$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

4930 026316 060037 001122          ADD      R0,$B0ADR          ;ADD RH11 BASE ADDRESS
4931 026322 012737 100000 001124    MOV      #ATA,$GDDAT        ;WHAT REGISTER SHOULD BE
4932 026330 013737 001126 001164    MOV      $BDDAT,$TMP0       ;MOVE REGISTER CONTENTS TO 'STMP0'
4933 026336 042737 077777 001164    BIC      #1CATA,$TMP0       ;SAVE SPECIFIED BITS
4934 026344 023737 001124 001164    CMP      $GDDAT,$TMP0       ;COMPARE THE BITS
4935 026352 001414          BEQ      70$                ;BR IF OK
4936 026354 013737 001126 001174    MOV      $BDDAT,$TMP4       ;COPY 'BAD DATA'
4937 026362 042737 100000 001174    BIC      #ATA,$TMP4         ;CLEAR THE MASKED BITS
4938 026370 053737 001174 001124    BIS      $TMP4,$GDDAT       ;'OR' WITH GOOD DATA FOR TYPEOUT
4939 026376 104016          ERROR   16                 ;TYPE MESSAGE 16
4940 026400 005137 001244          COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
4941 026404 000240          NOP
70$:
;*****
;RELEASE THE DRIVE FROM PORT A
4947 026406 113760 001224 000010    MOV      PORTA,RPCS2(R0)     ;SELECT PORT A
4948 026414 013737 001224 001234    MOV      PORTA,PTNBR         ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4949 026422 012760 000013 000000    MOV      #13,RPCS1(R0)      ;ISSUE RELEASE THROUGH PORT A
;VERIFY THAT THE DRIVE IS IN NEUTRAL
4953 026430 005037 001250          CLR      RELERR              ;CLEAR THE 'RELEASE ERROR' INDICATOR
4954 026434 012737 000012 001122    MOV      #RPDS1,$B0ADR       ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4955 026442 060037 001122          ADD      R0,$B0ADR          ;ADD THE I/O BASE ADDRESS
4956 026446 012737 011700 001124    MOV      #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
4957 026454 113760 001224 000010    MOV      PORTA,RPCS2(R0)     ;SELECT PORT A.
4958 026462 016037 000012 001170    MOV      RPDS1(R0),$TMP2     ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4959 026470 013737 001170 001164    MOV      $TMP2,$TMP0         ;COPY IT INTO 'STMP0'
4960 026476 042737 100100 001164    BIC      #ATA:VV,$TMP0       ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4961 026504 113760 001226 000010    MOV      PORTB,RPCS2(R0)     ;SELECT PORT B.
4962 026512 016037 000012 001172    MOV      RPDS1(R0),$TMP3     ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4963 026520 013737 001172 001166    MOV      $TMP3,$TMP1         ;COPY IT INTO 'STMP1'
4964 026526 042737 100100 001166    BIC      #ATA:VV,$TMP1       ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4965 026534 023737 001164 001166    CMP      $TMP0,$TMP1         ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4966 026542 001006          BNE      72$                ;BR IF NOT
4967 026544 005737 001164          TST      $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4968 026550 001045          BNE      74$                ;BR IF NOT
4969 026552 104046          ERROR   46                 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4970 026554 000137 026740          JMP      76$                ;BYPASS THE REST OF THE CHECKS
4971 026560 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT        ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4972 026566 013737 001226 001234    MOV      PORTB,PTNBR         ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4973 026574 113760 001226 000010    MOV      PORTB,RPCS2(R0)     ;SELECT PORT B.
4974 026602 005737 001164          TST      $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
4975 026606 001414          BEQ      73$                ;BR IF ZERO
4976 026610 013737 001224 001234    MOV      PORTA,PTNBR         ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4977 026616 013737 001172 001126    MOV      $TMP3,$BDDAT        ;'BAD DATA' FOR ERROR TYPE OUT
4978 026624 113760 001224 000010    MOV      PORTA,RPCS2(R0)     ;SELECT PORT A.
4979 026632 005737 001166          TST      $TMP1              ;SEE IF STATUS EQ ZERO FROM PORT B.
4980 026636 001012          BNE      74$                ;BR IF NOT
4981 026640 012737 177777 001250 73$:  MOV      #-1,RELERR          ;SET 'RELEASE ERROR' INDICATOR
4982 026646 012760 000011 000000    MOV      #11,RPCS1(R0)       ;CLEAR THE DRIVE
4983 026654 012760 000013 000000    MOV      #13,RPCS1(R0)       ;RELEASE THE DRIVE
4984 026662 104026          ERROR   26                 ;TYPE ERROR MESSAGE 26
4985 026664 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT        ;LOOK FOR BIT FAILURES WHEN RPDS1 READ

```

4986	026672	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
4987	026700	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
4988	026706	001401				BEQ	75\$;BR IF OK FROM PORT A.
4989	026710	104007				ERROR	?	;REPORT ERROR
4990	026712	013737	001172	001126	75\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4991	026720	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
4992	026726	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
4993	026734	001401				BEQ	76\$;BR IF OK
4994	026736	104007				ERROR	?	;REPORT ERROR
4995	026740	000240			76\$:	NOP		
4996	026742	000004			1\$:	SCOPE		;LOOP ?

4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052

026744
026744 005737 001274
026750 001406
026752 100002
026754 000137 002602
026760 012737 177777 001274
026766 112737 000022 001102
026774 012737 027016 001106
027002 012737 027016 001110
027010 012737 007640 001176
027016 012706 001100

027022 113760 001224 000010
027030 005060 000012
027034 012760 000011 000000
027042 012760 000013 000000
027050 113760 001226 000010
027056 005060 000012
027062 012760 000011 000000
027070 012760 000013 000000

027076 113760 001224 000010
027104 013737 001224 001236
027112 005060 000012
027116 013737 001226 001240
027124 113760 001226 000010
027132 013737 001226 001234

```
*****  
:TEST 22 TEST PORT REQUEST FROM PORT 'B'  
:VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE  
: DRIVE IS SEIZED BY THE OTHER PORT.  
: A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.  
: B. WRITE 0'S INTO RPDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL  
: SEIZED BY PORT 'A'.  
: C. ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE  
: SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR  
: PORT 'B' AND IS NOT SET FOR PORT 'A'.  
: D. ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE  
: RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.  
*****  
TST22: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?  
BEQ 2$ ;BR IF NOT  
BPL 1$ ;BR IF JUST ENTERED TEST  
JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
1$: MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR  
2$: MOV #22,$STSTM ;TEST NUMBER  
MOV #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS  
MOV #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS  
MOV #4000,$TIMES ;DO 4000. ITERATIONS  
TEST22: MOV #STACK,$SP ;LOAD THE STACK POINTER  
  
;CLEAR ATTENTION BITS FOR BOTH PORTS  
MOVB PORTA,RPCS2(RO) ;SELECT PORT #A  
CLR RPDS1(RO) ;SEIZE THE DRIVE  
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR  
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE  
MOVB PORTB,RPCS2(RO) ;SELECT PORT #B  
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'  
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR  
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE  
  
:*****  
;SEIZE THE DRIVE THROUGH PORT A  
MOVB PORTA,RPCS2(RO) ;SELECT PORT A  
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS  
CLR RPDS1(RO) ;WRITE RPDS1  
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS  
MOVB PORTB,RPCS2(RO) ;SELECT PORT B  
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
:*****
```

```

5053 ;SET PORT REQUEST
5054
5055 027140 005060 000012 CLR RPDS1(RO) ;SET PORT REQUEST FOR PORT B
5056
5057 ;*****
5058 ;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.
5059
5060
5061 ;RELEASE THE DRIVE FROM PORT A
5062
5063 027144 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5064 027152 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5065 027160 012760 000013 000000 MOV #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A
5066
5067 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5068
5069 027166 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5070 027172 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
5071 027200 012737 000012 001122 MOV #RPDS1,$BDAOR ;REGISTER ADDRESS INCREMENT
5072 027206 060037 001122 ADD RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
5073 027212 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5074 027220 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5075 027226 016037 000012 001164 MOV RPDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT B
5076 027234 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5077 027242 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5078 027250 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
5079 027256 001404 BEQ 66$ ;BR IF STATUS FROM PORT A ZERO
5080 027260 005737 001164 TST $TMPD ;IS STATUS FROM PORT B ZERO?
5081 027264 001401 BEQ 66$ ;BR IF ZERO
5082 027266 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
5083 027270 013737 001164 001126 66$: MOV $TMPD,$BDDAT ;CHECK STATUS FROM PORT B
5084 027276 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5085 027304 023737 001124 001126 CMP SGDDAT,$BDDAT ;COMPARE WITH CONSTANT
5086 027312 001401 BEQ 67$ ;BR IF OK
5087 027314 104027 ERROR 27 ;REPORT REGISTER ERROR
5088 027316 000240 67$: NOP
5089 027320 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5090 027326 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5091 027334 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5092 027340 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
5093 027346 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5094 027354 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
5095 027360 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
5096 027364 013737 001126 001164 MOV $BDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
5097 027372 042737 077777 001164 BIC #!CATA,$TMPD ;SAVE SPECIFIED BITS
5098 027400 023737 001124 001164 CMP SGDDAT,$TMPD ;COMPARE THE BITS
5099 027406 001414 BEQ 68$ ;BR IF OK
5100 027410 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
5101 027416 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
5102 027424 053737 001174 001124 BIS $TMP4,$SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5103 027432 104016 ERROR 16 ;TYPE MESSAGE 16
5104 027434 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5105 027440 000240 68$: NOP
5106 027442 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5107 027450 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5108 027456 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR

```



```

5109 027462 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
5110 027470 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5111 027476 060037 001122 ADD RO,SBADR ;ADD RHI1 BASE ADDRESS
5112 027502 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
5113 027510 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5114 027516 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
5115 027524 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5116 027532 001414 BEQ 70$ ;BR IF OK
5117 027534 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
5118 027542 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
5119 027550 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5120 027556 104016 ERROR 16 ;TYPE MESSAGE 16
5121 027560 005137 001244 COM 16 ;SET THE REGISTER COMPARE ERROR INDICATOR
5122 027564 000240 70$: NOP
5123
5124 ;*****
5125
5126 ;RELEASE THE DRIVE FROM PORT B
5127
5128 027566 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5129 027574 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5130 027602 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5131
5132 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5133
5134 027610 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5135 027614 012737 000012 001122 MOV #RPDS1,SBADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5136 027622 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
5137 027626 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5138 027634 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5139 027642 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5140 027650 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
5141 027656 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5142 027664 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
5143 027672 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5144 027700 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
5145 027706 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5146 027714 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5147 027722 001006 BNE 72$ ;BR IF NOT
5148 027724 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5149 027730 001045 BNE 74$ ;BR IF NOT
5150 027732 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5151 027734 000137 030120 JMP 76$ ;BYPASS THE REST OF THE CHECKS
5152 027740 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5153 027746 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5154 027754 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
5155 027762 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5156 027766 001414 BEQ 73$ ;BR IF ZERO
5157 027770 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5158 027776 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
5159 030004 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5160 030012 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5161 030016 001012 BNE 74$ ;BR IF NOT
5162 030020 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5163 030026 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
5164 030034 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE

```

F08

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 96
CZRJEB.P11 04-NOV-77 13:27

T22 TEST PORT REQUEST FROM PORT 'B'

SEQ 0096

```

S165 030042 104026          ERROR 26          ;TYPE ERROR MESSAGE 26
S166 030044 013737 001170 001126 74$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
S167 030052 013737 001224 001234      MOV PORTA,PTNBR ;CHANGE PORT NUMBER
S168 030060 023737 001124 001170      CMP $GDDAT,$STMP2 ;ALL BITS OK ?
S169 030066 001401          BEQ 75$          ;BR IF OK FROM PORT A.
S170 030070 104007          ERROR 7           ;REPORT ERROR
S171 030072 013737 001172 001126 75$: MOV STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
S172 030100 013737 001226 001234      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
S173 030106 023737 001124 001172      CMP $GDDAT,$STMP3 ;SEE IF READ OK FROM PORT B.
S174 030114 001401          BEQ 76$          ;BR IF OK
S175 030116 104007          ERROR 7           ;REPORT ERROR
S176 030120 000240          76$: NOP
S177 030122 000004          1$: SCOPE ;LOOP ?

```

```

S180 *****
S181 *TEST 23 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
S182 *
S183 *VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
S184 * REQUEST'.
S185 *
S186 * A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1. VERIFY THAT
S187 * THE DRIVE HAS BEEN SEIZED.
S188 *
S189 * B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT
S190 * SET.
S191 *
S192 * C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
S193 * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
S194 *
S195 *****

```

```

S196 030124          TST23: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
S197 030124 005737 001274      BEQ 2$          ;BR IF NOT
S198 030130 001406          BPL 1$          ;BR IF JUST ENTERED TEST
S199 030132 100002          JMP EXEC        ;RETURN & GET NEXT TEST NUMBER
S200 030134 000137 002602      1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
S201 030140 012737 177777 001274 2$: MOVB #23,$STSTN ;TEST NUMBER
S202 030146 112737 000023 001102      MOV #TEST23,$LPADR ;LOAD LOOP ON TEST ADDRESS
S203 030154 012737 030176 001106      MOV #TEST23,$LPERR ;LOAD LOOP ON ERROR ADDRESS
S204 030162 012737 030176 001110      MOV #4000,$TIMES ;DO 4000. ITERATIONS
S205 030170 012737 007640 001176      TEST23: MOV #STACK,SP ;LOAD THE STACK POINTER
S206 030176 012706 001100
S207
S208 ;CLEAR ATTENTION BITS FOR BOTH PORTS
S209
S210 030202 113760 001224 000010      MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
S211 030210 005060 000012          CLR RPDS1(RO) ;SEIZE THE DRIVE
S212 030214 012760 000011 000000      MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
S213 030222 012760 000013 000000      MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
S214 030230 113760 001226 000010      MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
S215 030236 005060 000012          CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
S216 030242 012760 000011 000000      MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
S217 030250 012760 000013 000000      MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
S218
S219
S220

```

```

5221 ;SEIZE THE DRIVE THROUGH PORT B
5222
5223 030256 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5224 030264 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
5225 030272 005760 000000 TST RPCS1(RO) ;READ RHCS1
5226 030276 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5227 030304 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5228 030312 013737 001224 001240 MOV PORTA,OPPR ;'OPPOSITE' PORT ADDRESS
5229 030320 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
5230 030326 010037 001122 MOV RO,$BDAADR ;RH11 BASE ADDRESS
5231 030332 062737 000012 001122 ADD #RPDS1,$BDAADR ;GENERATE BAD REGISTER ADDRESS
5232 030340 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
5233 030344 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
5234 030352 001403 BEQ 64$ ;BR IF IT IS
5235 030354 104004 ERROR 4 ;REPORT THE ERROR
5236 030356 000137 031132 JMP 1$ ;BYPASS REST OF THE SUBTEST
5237 030362
5238 030362 113760 001226 000010 64$: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5239 030370 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5240 030376 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5241 030404 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5242 030412 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5243 030420 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
5244 030424 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
5245 030432 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
5246 030440 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5247 030446 001401 BEQ 65$ ;BR IF THEY ARE
5248 030450 104005 ERROR 5 ;REPORT THE ERROR
5249 030452 000240
5250 030454 113760 001224 000010 65$: NOP
5251 030462 013737 001224 001234 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5252 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5253
5254 ;*****
5255 ;READ RPCS1 THROUGH PORT A - TRY TO SET PORT REQUEST
5256
5257 030470 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5258 030474 016037 000000 001126 MOV RPCS1(RO),$BDDAT ;GET CONTENTS OF RPCS1
5259 030502 012737 000000 001122 MOV #RPCS1,$BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5260 030510 060037 001122 ADD RO,$BDAADR ;ADD RH11 BASE ADDRESS
5261 030514 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5262 030520 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5263 030526 042737 173700 001164 BIC #1C4077,$TMP0 ;SAVE SPECIFIED BITS
5264 030534 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
5265 030542 001414 BEQ 66$ ;BR IF OK
5266 030544 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
5267 030552 042737 004077 001174 BIC #4077,$TMP4 ;CLEAR THE MASKED BITS
5268 030560 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5269 030566 104010 ERROR 10 ;REPORT THE ERROR
5270 030570 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5271 030574 000240 66$: NOP
5272
5273 ;*****
5274 ;DRIVE SHOULD RETURN TO NEUTRAL
5275
5276 ;RELEASE THE DRIVE FROM PORT B

```

```

5277
5278 030576 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
5279 030604 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5280 030612 012760 000013 000000      MOV    #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5281
5282 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5283
5284 030620 005037 001250      CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5285 030624 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5286 030632 060037 001122      ADD    RO,$BDADR ;ADD THE I/O BASE ADDRESS
5287 030636 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5288 030644 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
5289 030652 016037 000012 001170      MOV    RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5290 030660 013737 001170 001164      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
5291 030666 042737 100100 001164      BIC    #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5292 030674 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
5293 030702 016037 000012 001172      MOV    RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5294 030710 013737 001172 001166      MOV    STMP3,STMP1 ;COPY IT INTO 'STMP1'
5295 030716 042737 100100 001166      BIC    #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5296 030724 023737 001164 001166      CMP    STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5297 030732 001006      BNE    68$ ;BR IF NOT
5298 030734 005737 001164      TST    STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5299 030740 001045      BNE    70$ ;BR IF NOT
5300 030742 104046      ERROR  46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5301 030744 000137 031130      JMP    72$ ;BYPASS THE REST OF THE CHECKS
5302 030750 013737 001170 001126 68$:      MOV    STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5303 030756 013737 001226 001234      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5304 030764 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
5305 030772 005737 001164      TST    STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5306 030776 001414      BEQ    69$ ;BR IF ZERO
5307 031000 013737 001224 001234      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5308 031006 013737 001172 001126      MOV    STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
5309 031014 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
5310 031022 005737 001166      TST    STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5311 031026 001012      BNE    70$ ;BR IF NOT
5312 031030 012737 177777 001250 69$:      MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5313 031036 012760 000011 000000      MOV    #11,RPCS1(RO) ;CLEAR THE DRIVE
5314 031044 012760 000013 000000      MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
5315 031052 104026      ERROR  26 ;TYPE ERROR MESSAGE 26
5316 031054 013737 001170 001126 70$:      MOV    STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5317 031062 013737 001224 001234      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
5318 031070 023737 001124 001170      CMP    $GDDAT,STMP2 ;ALL BITS OK ?
5319 031076 001401      BEQ    71$ ;BR IF OK FROM PORT A.
5320 031100 104007      ERROR  7 ;REPORT ERROR
5321 031102 013737 001172 001126 71$:      MOV    STMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5322 031110 013737 001226 001234      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
5323 031116 023737 001124 001172      CMP    $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
5324 031124 001401      BEQ    72$ ;BR IF OK
5325 031126 104007      ERROR  7 ;REPORT ERROR
5326 031130 000240 72$:      NOP
5327 031132 000004 1$:      SCOPE ;LOOP ?
5328
5329 ;*****
5330 ;*TEST 24 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'
5331 ;*
5332 ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT

```

5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388

- ```

;* REQUEST'.
;*
;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1. VERIFY THAT
;* THE DRIVE HAS BEEN SEIZED.
;*
;* B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT
;* SET.
;*
;* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

```

\*\*\*\*\*

```

†ST24: 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$: MOV #24,$TSTNM ;TEST NUMBER
 MOV #TEST24,$LPADR ;LOAD LOOP ON TEST ADDRESS
 MOV #TEST24,$LPERR ;LOAD LOOP ON ERROR ADDRESS
 MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST24: MOV #STACK,SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

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

```

\*\*\*\*\*

;SEIZE THE DRIVE THROUGH PORT A

```

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

```

```

64$: MOVB PORTA,RPCS2(RO) ;SELECT PORT A
 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

# JOB

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 100  
 CZRJEBO.P11 04-NOV-77 13:27 T24

TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'

SEQ 0100

```

5389 031406 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5390 031414 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
5391 031422 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
5392 031430 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
5393 031434 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
5394 031442 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
5395 031450 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5396 031456 001401 BEQ 65$;BR IF THEY ARE
5397 031460 104005 ERROR 5 ;REPORT THE ERROR
5398 031462 000240 NOP
5399 031464 113760 001226 000010 65$: MOVB PORTB, RPCS2(RO) ;SELECT PORT B
5400 031472 013737 001226 001234 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5401
5402 ;*****
5403 ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
5404
5405 031500 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5406 031504 016037 000000 001126 MOV RPCS1(RO), $BDDAT ;GET CONTENTS OF RPCS1
5407 031512 012737 000000 001122 MOV #RPCS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5408 031520 060037 001122 ADD RO, $BDAOR ;ADD RHI1 BASE ADDRESS
5409 031524 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5410 031530 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
5411 031536 042737 173700 001164 BIC #1C4077, $TMP0 ;SAVE SPECIFIED BITS
5412 031544 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
5413 031552 001414 BEQ 66$;BR IF OK
5414 031554 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
5415 031562 042737 004077 001174 BIC #4077, $TMP4 ;CLEAR THE MASKED BITS
5416 031570 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5417 031576 104010 ERROR 10 ;REPORT THE ERROR
5418 031600 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5419 031604 000240 NOP
5420
5421 ;*****
5422 ;DRIVE SHOULD RETURN TO NEUTRAL
5423
5424
5425 ;RELEASE THE DRIVE FROM PORT A
5426
5427 031606 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT A
5428 031614 013737 001224 001234 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5429 031622 012760 000013 000000 MOV #13, RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5430
5431 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5432
5433 031630 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5434 031634 012737 000012 001122 MOV #RPDS1, $BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5435 031642 060037 001122 ADD RO, $BDAOR ;ADD THE I/O BASE ADDRESS
5436 031646 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
5437 031654 113760 001224 000010 MOVB PORTA, RPCS2(RO) ;SELECT PORT A.
5438 031662 016037 000012 001170 MOV RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5439 031670 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
5440 031676 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5441 031704 113760 001226 000010 MOVB PORTB, RPCS2(RO) ;SELECT PORT B.
5442 031712 016037 000012 001172 MOV RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5443 031720 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
5444 031726 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```

```

5445 031734 023737 001164 001166 CMP $TMP0,$TMP1 ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5446 031742 001006 BNE 68$; BR IF NOT
5447 031744 005737 001164 TST $TMP0 ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
5448 031750 001045 BNE 70$; BR IF NOT
5449 031752 104046 ERROR 46 ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5450 031754 000137 032140 JMP 72$; BYPASS THE REST OF THE CHECKS
5451 031760 013737 001170 001126 68$: MOV $TMP2,$BDDAT ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5452 031766 013737 001226 001234 MOV PORTB,PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5453 031774 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ; SELECT PORT B.
5454 032002 005737 001164 TST $TMP0 ; SEE IF STATUS EQ 0 FROM PORT A.
5455 032006 001414 BEQ 69$; BR IF ZERO
5456 032010 013737 001224 001234 MOV PORTA,PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5457 032016 013737 001172 001126 MOV $TMP3,$BDDAT ; 'BAD DATA' FOR ERROR TYPE OUT
5458 032024 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ; SELECT PORT A.
5459 032032 005737 001166 TST $TMP1 ; SEE IF STATUS EQ ZERO FROM PORT B.
5460 032036 001012 BNE 70$; BR IF NOT
5461 032040 012737 177777 001250 69$: MOV #-1,RELEA ; SET 'RELEASE ERROR' INDICATOR
5462 032046 012760 000011 000000 MOV #11,RPCS1(RO) ; CLEAR THE DRIVE
5463 032054 012760 000013 000000 MOV #13,RPCS1(RO) ; RELEASE THE DRIVE
5464 032062 104026 ERROR 26 ; TYPE ERROR MESSAGE 26
5465 032064 013737 001170 001126 70$: MOV $TMP2,$BDDAT ; LOOK FOR BIT FAILURES WHEN RPDS1 READ
5466 032072 013737 001224 001234 MOV PORTA,PTNBR ; CHANGE PORT NUMBER
5467 032100 023737 001124 001170 CMP $GDDAT,$TMP2 ; ALL BITS OK ?
5468 032106 001401 BEQ 71$; BR IF OK FROM PORT A.
5469 032110 104007 ERROR 7 ; REPORT ERROR
5470 032112 013737 001172 001126 71$: MOV $TMP3,$BDDAT ; CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5471 032120 013737 001226 001234 MOV PORTB,PTNBR ; CHANGE PORT NUMBER
5472 032126 023737 001124 001172 CMP $GDDAT,$TMP3 ; SEE IF READ OK FROM PORT B.
5473 032134 001401 BEQ 72$; BR IF OK
5474 032136 104007 ERROR 7 ; REPORT ERROR
5475 032140 000240 72$: NOP
5476 032142 000004 1$: SCOPE ; LOOP ?

```

```

5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498 032144
5499 032144 005737 001274
5500 032150 001406

```

```

*TEST 25 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
*
*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
* IS SEIZED BY THE OTHER PORT.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
*
* C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
*
* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED
* TO PORT 'A'.
*
* E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED
* TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

†ST25: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
 BEQ 2$;BR IF NOT

```

```

5501 032152 100002 BPL 1$;BR IF JUST ENTERED TEST
5502 032154 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
5503 032160 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
5504 032166 112737 000025 001102 2$: MOVVB #25,$TSTNM ;TEST NUMBER
5505 032174 012737 032216 001106 MOV #TEST25,$LPADR ;LOAD LOOP ON TEST ADDRESS
5506 032202 012737 032216 001110 MOV #TEST25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5507 032210 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
5508 032216 012706 001100 TEST25: MOV #STACK,$SP ;LOAD THE STACK POINTER
5509
5510 ;CLEAR ATTENTION BITS FOR BOTH PORTS
5511
5512 032222 113760 001224 000010 MOVVB PORTA,$RPCS2($R0) ;SELECT PORT #A
5513 032230 005060 000012 CLR $RPDS1($R0) ;SEIZE THE DRIVE
5514 032234 012760 000011 000000 MOV #11,$RPCS1($R0) ;ISSUE DRIVE CLEAR
5515 032242 012760 000013 000000 MOV #13,$RPCS1($R0) ;RELEASE THE DRIVE
5516 032250 113760 001226 000010 MOVVB PORTB,$RPCS2($R0) ;SELECT PORT #B
5517 032256 005060 000012 CLR $RPDS1($R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
5518 032262 012760 000011 000000 MOV #11,$RPCS1($R0) ;ISSUE DRIVE CLEAR
5519 032270 012760 000013 000000 MOV #13,$RPCS1($R0) ;RELEASE THE DRIVE
5520
5521 ;*****
5522
5523 ;SEIZE THE DRIVE THROUGH PORT B
5524
5525 032276 113760 001226 000010 MOVVB PORTB,$RPCS2($R0) ;SELECT PORT B
5526 032304 013737 001226 001236 MOV PORTB,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
5527 032312 005060 000012 CLR $RPDS1($R0) ;WRITE $RPDS1
5528 032316 113760 001224 000010 MOVVB PORTA,$RPCS2($R0) ;SELECT PORT A
5529 032324 013737 001224 001234 MOV PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5530 032332 013737 001224 001240 MOV PORTA,$OPPR ;'OPPOSITE' PORT ADDRESS
5531 032340 016037 000012 001126 MOV $RPDS1($R0),$SBDAT ;SEE IF DRIVE SEIZED BY PORT B
5532 032346 010037 001122 MOV $R0,$SBDADR ;$R0 IS BASE ADDRESS
5533 032352 062737 000012 001122 ADD #,$RPDS1,$SBDADR ;GENERATE BAD REGISTER ADDRESS
5534 032360 005037 001124 CLR $SGDAT ;REGISTER SHOULD BE ZERO
5535 032364 023737 001124 001126 CMP $SGDAT,$SBDAT ;IS THE REGISTER ZERO
5536 032372 001403 BEQ 64$;BR IF IT IS
5537 032374 104004 ERROR 4 ;REPORT THE ERROR
5538 032376 000137 033350 JMP 1$;BYPASS REST OF THE SUBTEST
5539 032402
5540 032402 113760 001226 000010 64$: MOVVB PORTB,$RPCS2($R0) ;SELECT PORT B
5541 032410 013737 001226 001234 MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5542 032416 016037 000012 001126 MOV $RPDS1($R0),$SBDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5543 032424 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$SGDAT ;EXPECTED STATUS
5544 032432 013737 001124 001166 MOV $SGDAT,$STMP1 ;USE GOOD DATA AS A MASK
5545 032440 005137 001166 COM $STMP1 ;COMPLEMENT THE EXPECTED STATUS
5546 032444 013737 001126 001164 MOV $SBDAT,$STMP0 ;SAVE THE ACTUAL STATUS
5547 032452 043737 001166 001164 BIC $STMP1,$STMP0 ;CLEAR UNWANTED BITS
5548 032460 023737 001124 001164 CMP $SGDAT,$STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5549 032466 001401 BEQ 65$;BR IF THEY ARE
5550 032470 104005 ERROR 5 ;REPORT THE ERROR
5551 032472 000240 65$: NOP
5552
5553 ;*****
5554 ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
5555
5556 032474 113760 001224 000010 MOVVB PORTA,$RPCS2($R0) ;SELECT PORT A

```



```

5557 032502 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5558 032510 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT A
5559
5560 ;*****
5561 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
5562
5563 032516 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5564 032522 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
5565 032530 012737 000012 001122 MOV #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5566 032536 060037 001122 ADD RO,SBDAOR ;ADD RHI1 BASE ADDRESS
5567 032542 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5568 032546 023737 001124 001126 CMP $GDDAT,SBDDAT ;IS THE REGISTER OK ?
5569 032554 001403 BEQ 66$;BR IF OK
5570 032556 104010 ERROR 10 ;REPORT THE ERROR
5571 032560 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5572 032564 016037 000000 001126 66$: MOV RPCS1(RO),SBDDAT ;GET THE CONTENTS OF RHCS1
5573 032572 012737 000000 001122 MOV #RPCS1,SBDAOR ;FORM ADDRESS OF REGISTER
5574 032600 060037 001122 ADD RO,SBDAOR ;ADDRESS BASE
5575 032604 032737 020000 001126 BIT #MCPE,SBDDAT ;IS 'MCPE' SET ?
5576 032612 001404 BEQ 67$;BR IF NOT
5577 032614 104011 ERROR 11 ;REPORT THE ERROR
5578 032616 012760 040000 000000 MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
5579 032624 000240 NOP
5580 032626 005737 001244 TST CKERR ;WAS RPDS1 NON ZERO ?
5581 032632 001402 BEQ +6 ;CONTENTS OF RPDS1 SEEN BY PORT A
5582 032634 000137 033350 JMP 1$;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5583
5584 ;*****
5585
5586 ;RELEASE THE DRIVE FROM PORT B
5587
5588 032640 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5589 032646 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5590 032654 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5591
5592 ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5593
5594 032662 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5595 032666 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5596 032674 012737 000012 001122 MOV #RPDS1,SBDAOR ;REGISTER ADDRESS INCREMENT
5597 032702 060037 001122 ADD RO,SBDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
5598 032706 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5599 032714 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5600 032722 016037 000012 001164 MOV RPDS1(RO),$TMPO ;READ STATUS REGISTER FROM PORT A
5601 032730 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
5602 032736 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5603 032744 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;DRIVE STATUS FROM PORT B
5604 032752 001404 BEQ 68$;BR IF STATUS FROM PORT B ZERO
5605 032754 005737 001164 TST $TMPO ;IS STATUS FROM PORT A ZERO ?
5606 032760 001401 BEQ 68$;BR IF ZERO
5607 032762 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
5608 032764 013737 001164 001126 68$: MOV $TMPO,SBDDAT ;CHECK STATUS FROM PORT A
5609 032772 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5610 033000 023737 001124 001126 CMP $GDDAT,SBDDAT ;COMPARE WITH CONSTANT
5611 033006 001401 BEQ 69$;BR IF OK
5612 033010 104027 ERROR 27 ;REPORT REGISTER ERROR

```

```

5613 033012 000240 69$: NOP
5614
5615 ;RELEASE THE DRIVE FROM PORT A
5616
5617 033014 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
5618 033022 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5619 033030 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5620
5621 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5622
5623 033036 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5624 033042 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5625 033050 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
5626 033054 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5627 033062 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5628 033070 016037 000012 001170 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5629 033076 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
5630 033104 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5631 033112 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
5632 033120 016037 000012 001172 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5633 033126 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
5634 033134 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5635 033142 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5636 033150 001006 BNE 70$;BR IF NOT
5637 033152 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5638 033156 001045 BNE 72$;BR IF NOT
5639 033160 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5640 033162 000137 033346 JMP 74$;BYPASS THE REST OF THE CHECKS
5641 033166 013737 001170 001126 70$: MOV STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5642 033174 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5643 033202 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
5644 033210 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5645 033214 001414 BEQ 71$;BR IF ZERO
5646 033216 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5647 033224 013737 001172 001126 MOV STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
5648 033232 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5649 033240 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5650 033244 001012 BNE 72$;BR IF NOT
5651 033246 012737 177777 001250 71$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5652 033254 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
5653 033262 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
5654 033270 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5655 033272 013737 001170 001126 72$: MOV STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5656 033300 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5657 033306 023737 001124 001170 CMP $GDDAT,STMP2 ;ALL BITS OK ?
5658 033314 001401 BEQ 73$;BR IF OK FROM PORT A.
5659 033316 104007 ERROR 7 ;REPORT ERROR
5660 033320 013737 001172 001126 73$: MOV STMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5661 033326 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5662 033334 023737 001124 001172 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
5663 033342 001401 BEQ 74$;BR IF OK
5664 033344 104007 ERROR 7 ;REPORT ERROR
5665 033346 000240 74$: NOP
5666 033350 000004 1$: SCOPE ;LOOP ?
5667
5668

```

::\*\*\*\*\*

5669  
5670  
5671  
5672  
5673  
5674  
5675  
5676  
5677  
5678  
5679  
5680  
5681  
5682  
5683  
5684  
5685  
5686  
5687 033352  
5688 033352 005737 001274  
5689 033356 001406  
5690 033360 100002  
5691 033362 000137 002602  
5692 033366 012737 177777 001274  
5693 033374 112737 000026 001102  
5694 033402 012737 033424 001106  
5695 033410 012737 033424 001110  
5696 033416 012737 007640 001176  
5697 033424 012706 001100  
5698  
5699  
5700  
5701 033430 113760 001224 000010  
5702 033436 005060 000012  
5703 033442 012760 000011 000000  
5704 033450 012760 000013 000000  
5705 033456 113760 001226 000010  
5706 033464 005060 000012  
5707 033470 012760 000011 000000  
5708 033476 012760 000013 000000  
5709  
5710  
5711  
5712  
5713  
5714 033504 113760 001224 000010  
5715 033512 013737 001224 001236  
5716 033520 005060 000012  
5717 033524 113760 001226 000010  
5718 033532 013737 001226 001234  
5719 033540 013737 001226 001240  
5720 033546 016037 000012 001126  
5721 033554 010037 001122  
5722 033560 062737 000012 001122  
5723 033566 005037 001124  
5724 033572 023737 001124 001126

```
;*TEST 26 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
;* IS SEIZED BY THE OTHER PORT.
;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
;* C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
;* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED
;* TO PORT 'B'.
;* E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED
;* TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
```

\*\*\*\*\*

```
†ST26: 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 #26,$TSTNM ;TEST NUMBER
MOV #TEST26,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST26,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST26: MOV #STACK,$SP ;LOAD THE STACK POINTER
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

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

;;\*\*\*\*\*

;SEIZE THE DRIVE THROUGH PORT A

```
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BDAADR ;R#11 BASE ADDRESS
ADD #RPDS1,$BDAADR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
```

```

5725 033600 001403 BEQ 64$;BR IF IT IS
5726 033602 104004 ERROR 4 ;REPORT THE ERROR
5727 033604 000137 034556 JMP 1$;BYPASS REST OF THE SUBTEST
5728 033610 64$:
5729 033610 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
5730 033616 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5731 033624 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5732 033632 012737 011700 001124 MOV #MOL!PGM!OPR!DRY!VV,SGDDAT ;EXPECTED STATUS
5733 033640 013737 001124 001166 MOV SGDDAT,$TMP1 ;USE GOOD DATA AS A MASK
5734 033646 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
5735 033652 013737 001126 001164 MOV SBDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
5736 033660 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
5737 033666 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
5738 033674 001401 BEQ 65$;BR IF THEY ARE
5739 033676 104005 ERROR 5 ;REPORT THE ERROR
5740 033700 000240 65$:
5741 NOP
5742 ;:*****
5743 ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
5744 ;:*****
5745 033702 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B
5746 033710 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5747 033716 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT B
5748 ;:*****
5749 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
5750 ;:*****
5751 ;:*****
5752 033724 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5753 033730 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
5754 033736 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5755 033744 060037 001122 ADD RO,SBADR ;ADD R111 BASE ADDRESS
5756 033750 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
5757 033754 023737 001124 001126 CMP $GDDAT,$SBDDAT ;IS THE REGISTER OK ?
5758 033762 001403 BEQ 66$;BR IF OK
5759 033764 104010 ERROR 10 ;REPORT THE ERROR
5760 033766 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5761 033772 016037 000000 001126 66$: MOV RPCS1(RO),SBDDAT ;GET THE CONTENTS OF RHCS1
5762 034000 012737 000000 001122 MOV #RPCS1,SBADR ;FORM ADDRESS OF REGISTER
5763 034006 060037 001122 ADD RO,SBADR ;ADDRESS BASE
5764 034012 032737 020000 001126 BIT #MCPE,SBDDAT ;IS 'MCPE' SET ?
5765 034020 001404 BEQ 67$;BR IF NOT
5766 034022 104011 ERROR 11 ;REPORT THE ERROR
5767 034024 012760 040000 000000 67$: MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
5768 034032 000240 NOP
5769 034034 005737 001244 TST CKERR ;WAS RPDS1 NON ZERO ?
5770 034040 001402 BEQ +6 ;CONTENTS OF RPDS1 SEEN BY PORT B
5771 034042 000137 034556 JMP 1$;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5772 ;:*****
5773 ;:*****
5774 ;RELEASE THE DRIVE FROM PORT A
5775 ;:*****
5776 ;:*****
5777 034046 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
5778 034054 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5779 034062 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
5780

```

```

5781 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5782
5783 034070 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
5784 034074 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5785 034102 012737 000012 001122 MOV #RPDS1,$BDDADR ;REGISTER ADDRESS INCREMENT
5786 034110 060037 001122 ADD RO,$BDDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
5787 034114 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B
5788 034122 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5789 034130 016037 000012 001164 MOV RPDS1(RO),STMP0 ;READ STATUS REGISTER FROM PORT B
5790 034136 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
5791 034144 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5792 034152 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
5793 034160 001404 BEQ 68$;BR IF STATUS FROM PORT A ZERO
5794 034162 005737 001164 TST STMP0 ;IS STATUS FROM PORT B ZERO ?
5795 034166 001401 BEQ 68$;BR IF ZERO
5796 034170 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
5797 034172 013737 001164 001126 68$: MOV STMP0,$BDDAT ;CHECK STATUS FROM PORT B
5798 034200 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
5799 034206 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
5800 034214 001401 BEQ 69$;BR IF OK
5801 034216 104027 ERROR 27 ;REPORT REGISTER ERROR
5802 034220 000240 69$: NOP
5803
5804 ;RELEASE THE DRIVE FROM PORT B
5805
5806 034222 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B
5807 034230 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5808 034236 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
5809
5810 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5811
5812 034244 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
5813 034250 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5814 034256 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
5815 034262 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
5816 034270 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
5817 034276 016037 000012 001170 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5818 034304 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
5819 034312 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5820 034320 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
5821 034326 016037 000012 001172 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5822 034334 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
5823 034342 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5824 034350 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5825 034356 001006 BNE 70$;BR IF NOT
5826 034360 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5827 034364 001045 BNE 72$;BR IF NOT
5828 034366 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5829 034370 000137 034554 JMP 74$;BYPASS THE REST OF THE CHECKS
5830 034374 013737 001170 001126 70$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5831 034402 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5832 034410 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
5833 034416 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
5834 034422 001414 BEQ 71$;BR IF ZERO
5835 034424 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5836 034432 013737 001172 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

```

5837 034440 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
5838 034446 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
5839 034452 001012 BNE 72$;BR IF NOT
5840 034454 012737 177777 001250 71$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
5841 034462 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
5842 034470 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
5843 034476 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
5844 034500 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5845 034506 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
5846 034514 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
5847 034522 001401 BEQ 73$;BR IF OK FROM PORT A.
5848 034524 104007 ERROR 7 ;REPORT ERROR
5849 034526 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5850 034534 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
5851 034542 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
5852 034550 001401 BEQ 74$;BR IF OK
5853 034552 104007 ERROR 7 ;REPORT ERROR
5854 034554 000240 74$: NOP
5855 034556 000004 1$: SCOPE ;LOOP ?

```

```

5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872

```

\*\*\*\*\*  
\*TEST 27 TEST SEIZE BY WRITING ATTENTION BIT  
\*  
\*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER  
\* (RPAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER  
\* PORT.  
\*  
\* A. WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; VERIFY THAT THE DRIVE  
\* IS SEIZED.  
\*  
\* B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE  
\* DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE  
\* OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.  
\*  
\*\*\*\*\*

```

5873 034560
5874 034560 005737 001274
5875 034564 001406
5876 034566 100002
5877 034570 000137 002602
5878 034574 012737 177777 001274 1$: MOV #-1,KYBCTL
5879 034602 112737 000027 001102 2$: MOV #27,$STSTM
5880 034610 012737 034632 001106 MOV #TEST27,$LPADR ;LOAD LOOP ON TEST ADDRESS
5881 034616 012737 034632 001110 MOV #TEST27,$LPERR ;LOAD LOOP ON ERROR ADDRESS
5882 034624 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
5883 034632 012706 001100 TEST27: MOV #STACK,$SP ;LOAD THE STACK POINTER
5884
5885
5886
5887
5888
5889
5890
5891
5892

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

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

```

```

5893 034676 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
5894 034704 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
5895
5896 ;:*****
5897 ;SELECT DRIVE OTHER THAN THAT BEING TESTED
5898
5899 034712 113760 001230 000010 MOV PORTC,RPCS2(RO) ;SELECT DRIVE NOT BEING TESTED
5900 034720 013737 001224 001236 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
5901
5902 ;:*****
5903 ;WRITE THE DRIVE'S ATTENTION BIT
5904
5905 034726 013760 001232 000016 MOV ASR1,RPAS(RO) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
5906 034734 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
5907 034742 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5908
5909 ;:*****
5910 ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
5911
5912 034750 005760 000012 TST RPDS1(RO) ;SEE THE REGISTER THROUGH PORT A ?
5913 034754 001014 BNE 1$;BR IF YES
5914 034756 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B
5915 034764 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5916 034772 005760 000012 TST RPDS1(RO) ;SEE REGISTER THROUGH PORT B ?
5917 034776 001021 BNE 2$;BR IF YES
5918 035000 104042 ERROR 42 ;DRIVE NOT IN NEUTRAL OR SEIZED
5919 035002 000137 036552 JMP 4$;BYPASS REST OF TEST
5920 035006
5921 035006 113760 001226 000010 1$: MOV PORTB,RPCS2(RO) ;SELECT PORT B
5922 035014 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5923 035022 005760 000012 TST RPDS1(RO) ;REGISTER SHOULD BE ZERO THROUGH PORT B
5924 035026 001002 BNE +6 ;BR IF STATUS REG IS NOT ZERO
5925 035030 000137 035702 JMP 3$;STATUS REG IS ZERO
5926 035034 104043 ERROR 43 ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
5927 035036 000137 036552 JMP 4$;BYPASS REST OF TEST
5928
5929 ;:*****
5930 ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
5931
5932 035042
5933 035042 005037 001244 2$: CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
5934 035046 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
5935 035054 012737 000012 001126 MOV #RPDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5936 035062 060037 001122 ADD RO,$BDDADR ;ADD RHI1 BASE ADDRESS
5937 035066 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
5938 035074 013737 001126 001164 MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
5939 035102 042737 106077 001164 BIC #71700,$STMP0 ;SAVE SPECIFIED BITS
5940 035110 023737 001124 001164 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
5941 035116 001414 BEQ 64$;BR IF OK
5942 035120 013737 001126 001174 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
5943 035126 042737 071700 001174 BIC #71700,$STMP4 ;CLEAR THE MASKED BITS
5944 035134 053737 001174 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
5945 035142 104010 ERROR 10 ;REPORT THE ERROR
5946 035144 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
5947 035150 000240
5948 035152 013737 001226 001236 64$: NOP
 MOV PORTB,SEIZPT ;ADDRESS FOR ERROR MESSAGE

```

|      |        |        |        |        |       |                                 |                                                                |
|------|--------|--------|--------|--------|-------|---------------------------------|----------------------------------------------------------------|
| 5949 | 035160 | 013737 | 001224 | 001240 | MOV   | PORTA,OPPRT                     | ;SAME AS ABOVE                                                 |
| 5950 |        |        |        |        |       |                                 |                                                                |
| 5951 |        |        |        |        |       |                                 | ;RELEASE THE DRIVE FROM PORT B                                 |
| 5952 |        |        |        |        |       |                                 |                                                                |
| 5953 | 035166 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                 | ;SELECT PORT B                                                 |
| 5954 | 035174 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 5955 | 035202 | 012760 | 000013 | 000000 | MOV   | #13,RPCS1(RO)                   | ;ISSUE RELEASE THROUGH PORT B                                  |
| 5956 |        |        |        |        |       |                                 |                                                                |
| 5957 |        |        |        |        |       |                                 | ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B |
| 5958 |        |        |        |        |       |                                 |                                                                |
| 5959 | 035210 | 005037 | 001250 |        | CLR   | RELERR                          | ;CLEAR 'RELEASE ERROR' INDICATOR                               |
| 5960 | 035214 | 012737 | 111700 | 001124 | MOV   | #ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT | ;COMPARISON CONSTANT                                           |
| 5961 | 035222 | 012737 | 000012 | 001122 | MOV   | #RPDS1,\$BDADR                  | ;REGISTER ADDRESS INCREMENT                                    |
| 5962 | 035230 | 060037 | 001122 |        | ADD   | RO,\$BDADR                      | ;REGISTER BASE ADDRESS FOR TYPEOUT                             |
| 5963 | 035234 | 113760 | 001224 | 000010 | MOVB  | PORTA,RPCS2(RO)                 | ;SELECT PORT A                                                 |
| 5964 | 035242 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 5965 | 035250 | 016037 | 000012 | 001164 | MOV   | RPDS1(RO),\$TMP0                | ;READ STATUS REGISTER FROM PORT A                              |
| 5966 | 035256 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                 | ;SELECT PORT B                                                 |
| 5967 | 035264 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 5968 | 035272 | 016037 | 000012 | 001126 | MOV   | RPDS1(RO),\$BDADR               | ;DRIVE STATUS FROM PORT B                                      |
| 5969 | 035300 | 001404 |        |        | BEQ   | 66\$                            | ;BR IF STATUS FROM PORT B ZERO                                 |
| 5970 | 035302 | 005737 | 001164 |        | TST   | \$TMP0                          | ;IS STATUS FROM PORT A ZERO ?                                  |
| 5971 | 035306 | 001401 |        |        | BEQ   | 66\$                            | ;BR IF ZERO                                                    |
| 5972 | 035310 | 104044 |        |        | ERROR | 44                              | ;REPORT DRIVE NOT SEIZED BY PORT A                             |
| 5973 | 035312 | 013737 | 001164 | 001126 | MOV   | \$TMP0,\$BDADR                  | ;CHECK STATUS FROM PORT A                                      |
| 5974 | 035320 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                     | ;CHANGE PORT ADDRESS FOR TYPEOUT                               |
| 5975 | 035326 | 023737 | 001124 | 001126 | CMP   | \$GDDAT,\$BDADR                 | ;COMPARE WITH CONSTANT                                         |
| 5976 | 035334 | 001401 |        |        | BEQ   | 67\$                            | ;BR IF OK                                                      |
| 5977 | 035336 | 104027 |        |        | ERROR | 27                              | ;REPORT REGISTER ERROR                                         |
| 5978 | 035340 | 000240 |        |        | NOP   |                                 |                                                                |
| 5979 |        |        |        |        |       |                                 |                                                                |
| 5980 |        |        |        |        |       |                                 | ;RELEASE THE DRIVE FROM PORT A                                 |
| 5981 |        |        |        |        |       |                                 |                                                                |
| 5982 | 035342 | 113760 | 001224 | 000010 | MOVB  | PORTA,RPCS2(RO)                 | ;SELECT PORT A                                                 |
| 5983 | 035350 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 5984 | 035356 | 012760 | 000013 | 000000 | MOV   | #13,RPCS1(RO)                   | ;ISSUE RELEASE THROUGH PORT A                                  |
| 5985 |        |        |        |        |       |                                 |                                                                |
| 5986 |        |        |        |        |       |                                 | ;VERIFY THAT THE DRIVE IS IN NEUTRAL                           |
| 5987 |        |        |        |        |       |                                 |                                                                |
| 5988 | 035364 | 005037 | 001250 |        | CLR   | RELERR                          | ;CLEAR THE 'RELEASE ERROR' INDICATOR                           |
| 5989 | 035370 | 012737 | 000012 | 001122 | MOV   | #RPDS1,\$BDADR                  | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT                         |
| 5990 | 035376 | 060037 | 001122 |        | ADD   | RO,\$BDADR                      | ;ADD THE I/O BASE ADDRESS                                      |
| 5991 | 035402 | 012737 | 011700 | 001124 | MOV   | #MOL!PGM!DPR!DRY!VV,\$GDDAT     | ;COMPARISON CONSTANT                                           |
| 5992 | 035410 | 113760 | 001224 | 000010 | MOVB  | PORTA,RPCS2(RO)                 | ;SELECT PORT A.                                                |
| 5993 | 035416 | 016037 | 000012 | 001170 | MOV   | RPDS1(RO),\$TMP2                | ;GET THE DRIVE STATUS REGISTER FROM PORT A.                    |
| 5994 | 035424 | 013737 | 001170 | 001164 | MOV   | \$TMP2,\$TMP0                   | ;COPY IT INTO 'TMP0'                                           |
| 5995 | 035432 | 042737 | 100100 | 001164 | BIC   | #ATA!VV,\$TMP0                  | ;CLEAR PORT DEPENDENT BITS FROM THE COPY                       |
| 5996 | 035440 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                 | ;SELECT PORT B.                                                |
| 5997 | 035446 | 016037 | 000012 | 001172 | MOV   | RPDS1(RO),\$TMP3                | ;GET THE DRIVE STATUS REGISTER FROM PORT B.                    |
| 5998 | 035454 | 013737 | 001172 | 001166 | MOV   | \$TMP3,\$TMP1                   | ;COPY IT INTO 'TMP1'                                           |
| 5999 | 035462 | 042737 | 100100 | 001166 | BIC   | #ATA!VV,\$TMP1                  | ;CLEAR PORT DEPENDENT BITS FROM THE COPY                       |
| 6000 | 035470 | 023737 | 001164 | 001166 | CMP   | \$TMP0,\$TMP1                   | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?             |
| 6001 | 035476 | 001006 |        |        | BNE   | 68\$                            | ;BR IF NOT                                                     |
| 6002 | 035500 | 005737 | 001164 |        | TST   | \$TMP0                          | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?                       |
| 6003 | 035504 | 001045 |        |        | BNE   | 70\$                            | ;BR IF NOT                                                     |
| 6004 | 035506 | 104046 |        |        | ERROR | 46                              | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED                     |



```

6005 035510 000137 035674 JMP 72$;BYPASS THE REST OF THE CHECKS
6006 035514 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6007 035522 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6008 035530 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
6009 035536 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6010 035542 001414 BEQ 69$;BR IF ZERO
6011 035544 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6012 035552 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6013 035560 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
6014 035566 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6015 035572 001012 BNE 70$;BR IF NOT
6016 035574 012737 177777 001250 69$: MOV #-1,RELEARR ;SET 'RELEASE ERROR' INDICATOR
6017 035602 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
6018 035610 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6019 035616 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6020 035620 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6021 035626 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6022 035634 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6023 035642 001401 BEQ 71$;BR IF OK FROM PORT A.
6024 035644 104007 ERROR 7 ;REPORT ERROR
6025 035646 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6026 035654 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6027 035662 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6028 035670 001401 BEQ 72$;BR IF OK
6029 035672 104007 ERROR 7 ;REPORT ERROR
6030 035674 000240 NOP
6031 035676 000137 036552 JMP 4$
6032
6033 ;*****
6034 ;THE DRIVE IS SEIZED BY PORT A. VERIFY THAT PORT B HAS PORT REQUEST SET
6035
6036 035702 3$:
6037 035702 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
6038 035710 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6039 035716 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6040 035722 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
6041 035730 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6042 035736 060037 001122 ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
6043 035742 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;WHAT REGISTER SHOULD BE
6044 035750 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6045 035756 042737 106077 001164 BIC #1C71700,$TMP0 ;SAVE SPECIFIED BITS
6046 035764 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6047 035772 001414 BEQ 73$;BR IF OK
6048 035774 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6049 036002 042737 071700 001174 BIC #71700,$TMP4 ;CLEAR THE MASKED BITS
6050 036010 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6051 036016 104010 ERROR 10 ;REPORT THE ERROR
6052 036020 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6053 036024 000240 NOP
6054 036026 013737 001224 001236 73$: MOV PORTA,SEIZPT ;ADDRESS FOR ERROR MESSAGE
6055 036034 013737 001226 001240 MOV PORTB,OPPRT ;SAME AS ABOVE
6056
6057 ;RELEASE THE DRIVE FROM PORT A
6058
6059 036042 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
6060 036050 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 112  
 CZRJEB.P11 04-NOV-77 13:27

T27 TEST SEIZE BY WRITING ATTENTION BIT

SEQ 0112

|      |        |        |        |        |       |       |                                |                                                                |
|------|--------|--------|--------|--------|-------|-------|--------------------------------|----------------------------------------------------------------|
| 6061 | 036056 | 012760 | 000013 | 000000 |       | MOV   | #13,RPCS1(RO)                  | ;ISSUE RELEASE THROUGH PORT A                                  |
| 6062 |        |        |        |        |       |       |                                |                                                                |
| 6063 |        |        |        |        |       |       |                                | ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A |
| 6064 |        |        |        |        |       |       |                                |                                                                |
| 6065 | 036064 | 005037 | 001250 |        |       | CLR   | RELEA                          | ;CLEAR 'RELEASE ERROR' INDICATOR                               |
| 6066 | 036070 | 012737 | 111700 | 001124 |       | MOV   | #ATA!MOL!PGM!DPR!DRY!VV,SGDDAT | ;COMPARISON CONSTANT                                           |
| 6067 | 036076 | 012737 | 000012 | 001122 |       | MOV   | #RPDS1,\$BDADR                 | ;REGISTER ADDRESS INCREMENT                                    |
| 6068 | 036104 | 060037 | 001122 |        |       | ADD   | RO,\$BDADR                     | ;REGISTER BASE ADDRESS FOR TYPEOUT                             |
| 6069 | 036110 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO)                | ;SELECT PORT B                                                 |
| 6070 | 036116 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR                    | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 6071 | 036124 | 016037 | 000012 | 001164 |       | MOV   | RPDS1(RO),STMP0                | ;READ STATUS REGISTER FROM PORT B                              |
| 6072 | 036132 | 113760 | 001224 | 000010 |       | MOVB  | PORTA,RPCS2(RO)                | ;SELECT PORT A                                                 |
| 6073 | 036140 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR                    | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 6074 | 036146 | 016037 | 000012 | 001126 |       | MOV   | RPDS1(RO),\$BDDAT              | ;DRIVE STATUS FROM PORT A                                      |
| 6075 | 036154 | 001404 |        |        |       | BEQ   | 75\$                           | ;BR IF STATUS FROM PORT A ZERO                                 |
| 6076 | 036156 | 005737 | 001164 |        |       | TST   | STMP0                          | ;IS STATUS FROM PORT B ZERO ?                                  |
| 6077 | 036162 | 001401 |        |        |       | BEQ   | 75\$                           | ;BR IF ZERO                                                    |
| 6078 | 036164 | 104044 |        |        |       | ERROR | 44                             | ;REPORT DRIVE NOT SEIZED BY PORT B                             |
| 6079 | 036166 | 013737 | 001164 | 001126 | 75\$: | MOV   | STMP0,\$BDDAT                  | ;CHECK STATUS FROM PORT B                                      |
| 6080 | 036174 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR                    | ;CHANGE PORT ADDRESS FOR TYPEOUT                               |
| 6081 | 036202 | 023737 | 001124 | 001126 |       | CMP   | SGDDAT,\$BDDAT                 | ;COMPARE WITH CONSTANT                                         |
| 6082 | 036210 | 001401 |        |        |       | BEQ   | 76\$                           | ;BR IF OK                                                      |
| 6083 | 036212 | 104027 |        |        |       | ERROR | 27                             | ;REPORT REGISTER ERROR                                         |
| 6084 | 036214 | 000240 |        |        | 76\$: | NOP   |                                |                                                                |
| 6085 |        |        |        |        |       |       |                                |                                                                |
| 6086 |        |        |        |        |       |       |                                | ;RELEASE THE DRIVE FROM PORT B                                 |
| 6087 |        |        |        |        |       |       |                                |                                                                |
| 6088 | 036216 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO)                | ;SELECT PORT B                                                 |
| 6089 | 036224 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR                    | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT                     |
| 6090 | 036232 | 012760 | 000013 | 000000 |       | MOV   | #13,RPCS1(RO)                  | ;ISSUE RELEASE THROUGH PORT B                                  |
| 6091 |        |        |        |        |       |       |                                |                                                                |
| 6092 |        |        |        |        |       |       |                                | ;VERIFY THAT THE DRIVE IS IN NEUTRAL                           |
| 6093 |        |        |        |        |       |       |                                |                                                                |
| 6094 | 036240 | 005037 | 001250 |        |       | CLR   | RELEA                          | ;CLEAR THE 'RELEASE ERROR' INDICATOR                           |
| 6095 | 036244 | 012737 | 000012 | 001122 |       | MOV   | #RPDS1,\$BDADR                 | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT                         |
| 6096 | 036252 | 060037 | 001122 |        |       | ADD   | RO,\$BDADR                     | ;ADD THE I/O BASE ADDRESS                                      |
| 6097 | 036256 | 012737 | 011700 | 001124 |       | MOV   | #MOL!PGM!DPR!DRY!VV,SGDDAT     | ;COMPARISON CONSTANT                                           |
| 6098 | 036264 | 113760 | 001224 | 000010 |       | MOVB  | PORTA,RPCS2(RO)                | ;SELECT PORT A.                                                |
| 6099 | 036272 | 016037 | 000012 | 001170 |       | MOV   | RPDS1(RO),STMP2                | ;GET THE DRIVE STATUS REGISTER FROM PORT A.                    |
| 6100 | 036300 | 013737 | 001170 | 001164 |       | MOV   | STMP2,STMP0                    | ;COPY IT INTO 'STMP0'                                          |
| 6101 | 036306 | 042737 | 100100 | 001164 |       | BIC   | #ATA!VV,STMP0                  | ;CLEAR PORT DEPENDENT BITS FROM THE COPY                       |
| 6102 | 036314 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO)                | ;SELECT PORT B.                                                |
| 6103 | 036322 | 016037 | 000012 | 001172 |       | MOV   | RPDS1(RO),STMP3                | ;GET THE DRIVE STATUS REGISTER FROM PORT B.                    |
| 6104 | 036330 | 013737 | 001172 | 001166 |       | MOV   | STMP3,STMP1                    | ;COPY IT INTO 'STMP1'                                          |
| 6105 | 036336 | 042737 | 100100 | 001166 |       | BIC   | #ATA!VV,STMP1                  | ;CLEAR PORT DEPENDENT BITS FROM THE COPY                       |
| 6106 | 036344 | 023737 | 001164 | 001166 |       | CMP   | STMP0,STMP1                    | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?             |
| 6107 | 036352 | 001006 |        |        |       | BNE   | 77\$                           | ;BR IF NOT                                                     |
| 6108 | 036354 | 005737 | 001164 |        |       | TST   | STMP0                          | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?                       |
| 6109 | 036360 | 001045 |        |        |       | BNE   | 79\$                           | ;BR IF NOT                                                     |
| 6110 | 036362 | 104046 |        |        |       | ERROR | 46                             | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED                     |
| 6111 | 036364 | 000137 | 036550 |        |       | JMP   | 81\$                           | ;BYPASS THE REST OF THE CHECKS                                 |
| 6112 | 036370 | 013737 | 001170 | 001126 | 77\$: | MOV   | STMP2,\$BDDAT                  | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE                    |
| 6113 | 036376 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR                    | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL               |
| 6114 | 036404 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO)                | ;SELECT PORT B.                                                |
| 6115 | 036412 | 005737 | 001164 |        |       | TST   | STMP0                          | ;SEE IF STATUS EQ 0 FROM PORT A.                               |
| 6116 | 036416 | 001414 |        |        |       | BEQ   | 78\$                           | ;BR IF ZERO                                                    |

```

6117 036420 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6118 036426 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6119 036434 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A.
6120 036442 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6121 036446 001012 BNE 79$;BR IF NOT
6122 036450 012737 177777 001250 78$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6123 036456 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
6124 036464 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6125 036472 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6126 036474 013737 001170 001126 79$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6127 036502 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6128 036510 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6129 036516 001401 BEQ 80$;BR IF OK FROM PORT A.
6130 036520 104007 ERROR 7 ;REPORT ERROR
6131 036522 013737 001172 001126 80$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6132 036530 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6133 036536 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6134 036544 001401 BEQ 81$;BR IF OK
6135 036546 104007 ERROR 7 ;REPORT ERROR
6136 036550 000240 81$: NOP
6137 036552 000004 4$: SCOPE ;LOOP ?

```

```

6140
6141 :*****
6142 :*TEST 30 TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
6143 :*
6144 :*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
6145 :* THE DRIVE'S ATTENTION BIT.
6146 :*
6147 :* A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
6148 :* BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
6149 :*
6150 :* B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
6151 :*
6152 :*****

```

```

6153 036554 005737 001274 1$: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6154 036560 001406 BEQ 2$;BR IF NOT
6155 036562 100002 BPL 1$;BR IF JUST ENTERED TEST
6156 036564 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6157 036570 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6158 036576 112737 000030 001102 2$: MOV #30,$STSTNM ;TEST NUMBER
6159 036604 012737 036626 001106 MOV #TEST30,$LPADR ;LOAD LOOP ON TEST ADDRESS
6160 036612 012737 036626 001110 MOV #TEST30,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6161 036620 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
6162 036626 012706 001100 TEST30: MOV #STACK,SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

6166 036632 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT #A
6167 036640 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
6168 036644 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6169 036652 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6170 036660 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT #B
6171 036666 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
6172 036672 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR

```

K09

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 114  
CZRJEB.P11 04-NOV-77 13:27 T30

TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT

SEQ 0114

```

6173 036700 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6174 036706 113760 001230 000010 MOV PORTC,RPCS2(RO) ;SELECT DRIVE NOT BEING TESTED
6175
6176 ;:*****
6177 ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST
6178
6179 036714 013737 001232 001164 MOV ASR1,$TMP0 ;STORE ATTN BIT FOR PORT A
6180 036722 005137 001164 COM $TMP0 ;COMPLEMENT IT
6181 036726 013760 001164 000016 MOV $TMP0,RPAS(RO) ;WRITE THE ATTN REGISTER
6182
6183 ;:*****
6184 ;VERIFY THAT DRIVE REMAINED IN NEUTRAL
6185
6186
6187 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6188
6189 036734 005037 001250 CLR RELEARR ;CLEAR THE 'RELEASE ERROR' INDICATOR
6190 036740 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6191 036746 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
6192 036752 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
6193 036760 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A.
6194 036766 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6195 036774 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
6196 037002 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6197 037010 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B.
6198 037016 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6199 037024 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
6200 037032 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6201 037040 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6202 037046 001006 BNE 64$;BR IF NOT
6203 037050 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6204 037054 001045 BNE 66$;BR IF NOT
6205 037056 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6206 037060 000137 037244 JMP 68$;BYPASS THE REST OF THE CHECKS
6207 037064 013737 001170 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6208 037072 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6209 037100 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B.
6210 037106 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
6211 037112 001414 BEQ 65$;BR IF ZERO
6212 037114 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6213 037122 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6214 037130 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A.
6215 037136 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6216 037142 001012 BNE 66$;BR IF NOT
6217 037144 012737 177777 001250 65$: MOV #-1,RELEARR ;SET 'RELEASE ERROR' INDICATOR
6218 037152 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
6219 037160 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6220 037166 104021 ERROR 21 ;TYPE ERROR MESSAGE 21
6221 037170 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6222 037176 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6223 037204 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6224 037212 001401 BEQ 67$;BR IF OK FROM PORT A.
6225 037214 104007 ERROR 7 ;REPORT ERROR
6226 037216 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6227 037224 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6228 037232 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.

```

6229 037240 001401  
6230 037242 104007  
6231 037244 000240  
6232 037246 000004

68S: BEQ 68S ;BR IF OK  
ERROR 7 ;REPORT ERROR  
NOP  
SCOPE ;LOOP ?

6233  
6234  
6235  
6236  
6237  
6238  
6239  
6240  
6241  
6242  
6243  
6244  
6245  
6246  
6247  
6248

```

*TEST 31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
*
*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
*
* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR
* PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

```

6249 037250  
6250 037250 005737 001274  
6251 037254 001406  
6252 037256 100002  
6253 037260 000137 002602  
6254 037264 012737 177777 001274  
6255 037272 112737 000031 001102  
6256 037300 012737 037322 001106  
6257 037306 012737 037322 001110  
6258 037314 012737 000004 001176  
6259 037322 012706 001100

```
TST31: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
 BEQ 2S ;BR IF NOT
 BPL 1S ;BR IF JUST ENTERED TEST
 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2S: MOVVB #31,$STSTNM ;TEST NUMBER
 MOV #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
 MOV #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
 MOV #4,$TIMES ;DO 4 ITERATIONS
TEST31: MOV #STACK,$SP ;LOAD THE STACK POINTER
```

6260  
6261  
6262  
6263 037326 113760 001224 000010  
6264 037334 005060 000012  
6265 037340 012760 000011 000000  
6266 037346 012760 000013 000000  
6267 037354 113760 001226 000010  
6268 037362 005060 000012  
6269 037366 012760 000011 000000  
6270 037374 012760 000013 000000

```
 ;CLEAR ATTENTION BITS FOR BOTH PORTS
 MOVVB PORTA,$RPCS2(R0) ;SELECT PORT #A
 CLR RPDS1(R0) ;SEIZE THE DRIVE
 MOV #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
 MOV #13,$RPCS1(R0) ;RELEASE THE DRIVE
 MOVVB PORTB,$RPCS2(R0) ;SELECT PORT #B
 CLR RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
 MOV #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
 MOV #13,$RPCS1(R0) ;RELEASE THE DRIVE
;*****
```

6271  
6272  
6273  
6274

```
 ;SEIZE THE DRIVE THROUGH PORT A
 MOVVB PORTA,$RPCS2(R0) ;SELECT PORT A
 MOV PORTA,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
 CLR RPDS1(R0) ;WRITE RPDS1
 MOV PORTB,$OPPRT ;'OPPOSITE' PORT ADDRESS
```

6275 037402 113760 001224 000010  
6276 037410 013737 001224 001236  
6277 037416 005060 000012  
6278 037422 013737 001226 001240  
6279  
6280  
6281  
6282  
6283 037430 012760 177777 000014  
6284

```

;FORCE AN ERROR
MOV #-1,$RPER1(R0) ;SET ERROR BITS
```

```

6285 ;:*****
6286 ;:START THE TIMER
6287
6288 037436 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
6289 037442 012737 003720 001254 MOV #2000, WATCH ;SET WATCH TO 2000 MS
6290 037450 113760 001226 000010 MOVVB PORTB,APCS2(RO) ;SELECT PORT B
6291 037456 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6292
6293 ;:*****
6294 ;:WAIT FOR DRIVE TO TIMEOUT
6295
6296 037464 005760 000012 1$: TST RPDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
6297 037470 001004 BNE 2$;BR IF DRIVE RELEASED
6298 037472 005737 001254 TST WATCH ;WATCH AT ZERO ?
6299 037476 001372 BNE 1$;BR IF NOT
6300 037500 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
6301 037502
6302 037502 113760 001224 000010 2$: MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
6303 037510 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6304
6305 ;:*****
6306 ;:THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET
6307
6308 037516 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6309 037522 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
6310 037530 012737 000012 001122 MOV #RPDS1, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6311 037536 060037 001122 ADD RO, $BDDADR ;ADD RH11 BASE ADDRESS
6312 037542 012737 040000 001124 MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
6313 037550 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6314 037556 042737 137777 001164 BIC #1C4000, $TMP0 ;SAVE SPECIFIED BITS
6315 037564 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6316 037572 001414 BEQ 66$;BR IF OK
6317 037574 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6318 037602 042737 040000 001174 BIC #40000, $TMP4 ;CLEAR THE MASKED BITS
6319 037610 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6320 037616 104023 ERROR 23 ;TYPE MESSAGE 23
6321 037620 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6322 037624 000240
6323 66$: NOP
6324
6325 ;:*****
6326 ;:THE ERROR REGISTER SHOULD CONTAIN 1'S
6327
6327 037626 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6328 037632 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
6329 037640 012737 000014 001122 MOV #RPER1, $BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6330 037646 060037 001122 ADD RO, $BDDADR ;ADD RH11 BASE ADDRESS
6331 037652 012737 177777 001124 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
6332 037660 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
6333 037666 001403 BEQ 68$;BR IF OK
6334 037670 104010 ERROR 10 ;REPORT THE ERROR
6335 037672 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6336 037676 000240
6337 68$: NOP
6338
6339 ;:*****
6340 ;:THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

```

|      |        |        |        |        |       |                  |                                                    |                                             |
|------|--------|--------|--------|--------|-------|------------------|----------------------------------------------------|---------------------------------------------|
| 6341 | 037700 | 005037 | 001244 |        | CLR   | CKERR            | ;CLEAR THE 'CHECK ERROR' INDICATOR                 |                                             |
| 6342 | 037704 | 016037 | 000012 | 001126 | MOV   | RPDS1(RO),SBDDAT | ;GET CONTENTS OF RPDS1                             |                                             |
| 6343 | 037712 | 012737 | 000012 | 001122 | MOV   | #RPDS1,SBADR     | ;FORM REGISTER ADDRESS OF ERROR MESSAGE            |                                             |
| 6344 | 037720 | 060037 | 001122 |        | ADD   | RO,SBADR         | ;ADD RH11 BASE ADDRESS                             |                                             |
| 6345 | 037724 | 012737 | 100000 | 001124 | MOV   | #ATA,\$GDDAT     | ;WHAT REGISTER SHOULD BE                           |                                             |
| 6346 | 037732 | 013737 | 001126 | 001164 | MOV   | SBDDAT,\$TMP0    | ;MOVE REGISTER CONTENTS TO '\$TMP0'                |                                             |
| 6347 | 037740 | 042737 | 077777 | 001164 | BIC   | #1CATA,\$TMP0    | ;SAVE SPECIFIED BITS                               |                                             |
| 6348 | 037746 | 023737 | 001124 | 001164 | CMP   | \$GDDAT,\$TMP0   | ;COMPARE THE BITS                                  |                                             |
| 6349 | 037754 | 001414 |        |        | BEG   | 70\$             | ;BR IF OK                                          |                                             |
| 6350 | 037756 | 013737 | 001126 | 001174 | MOV   | SBDDAT,\$TMP4    | ;COPY 'BAD DATA'                                   |                                             |
| 6351 | 037764 | 042737 | 100000 | 001174 | BIC   | #ATA,\$TMP4      | ;CLEAR THE MASKED BITS                             |                                             |
| 6352 | 037772 | 053737 | 001174 | 001124 | BIS   | \$TMP4,\$GDDAT   | ; 'OR' WITH GOOD DATA FOR TYPEOUT                  |                                             |
| 6353 | 040000 | 104041 |        |        | ERROR | 41               | ;TYPE MESSAGE 41                                   |                                             |
| 6354 | 040002 | 005137 | 001244 |        | COM   | CKERR            | ;SET THE REGISTER COMPARE ERROR INDICATOR          |                                             |
| 6355 | 040006 | 000240 |        |        | 70\$: | NOP              |                                                    |                                             |
| 6356 |        |        |        |        |       |                  |                                                    |                                             |
| 6357 |        |        |        |        |       |                  |                                                    |                                             |
| 6358 |        |        |        |        |       |                  |                                                    |                                             |
| 6359 |        |        |        |        |       |                  |                                                    |                                             |
| 6360 |        |        |        |        |       |                  |                                                    |                                             |
| 6361 |        |        |        |        |       |                  |                                                    |                                             |
| 6362 | 040010 | 005037 | 001250 |        | CLR   | RELERR           | ;CLEAR THE 'RELEASE ERROR' INDICATOR               |                                             |
| 6363 | 040014 | 012737 | 000012 | 001122 | MOV   | #RPDS1,SBADR     | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT             |                                             |
| 6364 | 040022 | 060037 | 001122 |        | ADD   | RO,SBADR         | ;ADD THE I/O BASE ADDRESS                          |                                             |
| 6365 | 040026 | 012737 | 051700 | 001124 | MOV   | #51700,\$GDDAT   | ;COMPARISON CONSTANT                               |                                             |
| 6366 | 040034 | 113760 | 001224 | 000010 | MOVB  | PORTA,APCS2(RO)  | ;SELECT PORT A.                                    |                                             |
| 6367 | 040042 | 016037 | 000012 | 001170 | MOV   | RPDS1(RO),\$TMP2 | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |                                             |
| 6368 | 040050 | 013737 | 001170 | 001164 | MOV   | \$TMP2,\$TMP0    | ;COPY IT INTO '\$TMP0'                             |                                             |
| 6369 | 040056 | 042737 | 100100 | 001164 | BIC   | #ATA!VV,\$TMP0   | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |                                             |
| 6370 | 040064 | 113760 | 001226 | 000010 | MOVB  | PORTB,APCS2(RO)  | ;SELECT PORT B.                                    |                                             |
| 6371 | 040072 | 016037 | 000012 | 001172 | MOV   | RPDS1(RO),\$TMP3 | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |                                             |
| 6372 | 040100 | 013737 | 001172 | 001166 | MOV   | \$TMP3,\$TMP1    | ;COPY IT INTO '\$TMP1'                             |                                             |
| 6373 | 040106 | 042737 | 100100 | 001166 | BIC   | #ATA!VV,\$TMP1   | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |                                             |
| 6374 | 040114 | 023737 | 001164 | 001166 | CMP   | \$TMP0,\$TMP1    | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |                                             |
| 6375 | 040122 | 001006 |        |        | BNE   | 72\$             | ;BR IF NOT                                         |                                             |
| 6376 | 040124 | 005737 | 001164 |        | TST   | \$TMP0           | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |                                             |
| 6377 | 040130 | 001045 |        |        | BNE   | 74\$             | ;BR IF NOT                                         |                                             |
| 6378 | 040132 | 104046 |        |        | ERROR | 46               | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |                                             |
| 6379 | 040134 | 000137 | 040334 |        | JMP   | 76\$             | ;BYPASS THE REST OF THE CHECKS                     |                                             |
| 6380 | 040140 | 013737 | 001170 | 001126 | 72\$: | MOV              | \$TMP2,\$BDDAT                                     | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE |
| 6381 | 040146 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR      | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |                                             |
| 6382 | 040154 | 113760 | 001226 | 000010 | MOVB  | PORTB,APCS2(RO)  | ;SELECT PORT B.                                    |                                             |
| 6383 | 040162 | 005737 | 001164 |        | TST   | \$TMP0           | ;SEE IF STATUS EQ 0 FROM PORT A.                   |                                             |
| 6384 | 040166 | 001414 |        |        | BEG   | 73\$             | ;BR IF ZERO                                        |                                             |
| 6385 | 040170 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR      | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |                                             |
| 6386 | 040176 | 013737 | 001172 | 001126 | MOV   | \$TMP3,\$BDDAT   | ; 'BAD DATA' FOR ERROR TYPE OUT                    |                                             |
| 6387 | 040204 | 113760 | 001224 | 000010 | MOVB  | PORTA,APCS2(RO)  | ;SELECT PORT A.                                    |                                             |
| 6388 | 040212 | 005737 | 001166 |        | TST   | \$TMP1           | ;SEE IF STATUS EQ ZERO FROM PORT B.                |                                             |
| 6389 | 040216 | 001012 |        |        | BNE   | 74\$             | ;BR IF NOT                                         |                                             |
| 6390 | 040220 | 012737 | 177777 | 001250 | 73\$: | MOV              | #-1,RELERR                                         | ;SET 'RELEASE ERROR' INDICATOR              |
| 6391 | 040226 | 012760 | 000011 | 000000 | MOV   | #11,APCS1(RO)    | ;CLEAR THE DRIVE                                   |                                             |
| 6392 | 040234 | 012760 | 000013 | 000000 | MOV   | #13,APCS1(RO)    | ;RELEASE THE DRIVE                                 |                                             |
| 6393 | 040242 | 104026 |        |        | ERROR | 26               | ;TYPE ERROR MESSAGE 26                             |                                             |
| 6394 | 040244 | 013737 | 001170 | 001126 | 74\$: | MOV              | \$TMP2,\$BDDAT                                     | ;LOOK FOR BIT FAILURES WHEN RPDS1 READ      |
| 6395 | 040252 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR      | ;CHANGE PORT NUMBER                                |                                             |
| 6396 | 040260 | 042737 | 100000 | 001170 | BIC   | #ATA,\$TMP2      | ;DON'T CHECK THE ATTN BIT                          |                                             |

::\*\*\*\*\*

;VERIFY THAT THE DRIVE IS IN NEUTRAL

|      |        |        |        |        |           |                 |                                               |
|------|--------|--------|--------|--------|-----------|-----------------|-----------------------------------------------|
| 6397 | 040266 | 023737 | 001124 | 001170 | CMP       | \$GDDAT,\$STMP2 | : ALL BITS OK ?                               |
| 6398 | 040274 | 001401 |        |        | BEQ       | 75\$            | : BR IF OK FROM PORT A.                       |
| 6399 | 040276 | 104007 |        |        | ERROR     | 7               | : REPORT ERROR                                |
| 6400 | 040300 | 013737 | 001172 | 001126 | 75\$: MOV | \$STMP3,\$SDDAT | : CHECK RPDS1 FOR BIT FAILURES - FROM PORT B. |
| 6401 | 040306 | 013737 | 001226 | 001234 | MOV       | PORTB,\$PTNBR   | : CHANGE PORT NUMBER                          |
| 6402 | 040314 | 042737 | 100000 | 001172 | BIC       | #ATA,\$STMP3    | : DON'T CHECK THE ATTN BIT                    |
| 6403 | 040322 | 023737 | 001124 | 001172 | CMP       | \$GDDAT,\$STMP3 | : SEE IF READ OK FROM PORT B.                 |
| 6404 | 040330 | 001401 |        |        | BEQ       | 76\$            | : BR IF OK                                    |
| 6405 | 040332 | 104007 |        |        | ERROR     | 7               | : REPORT ERROR                                |
| 6406 | 040334 | 000240 |        |        | 76\$: NOP |                 |                                               |

\*\*\*\*\*  
: THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

|      |        |        |        |        |           |                     |                                             |
|------|--------|--------|--------|--------|-----------|---------------------|---------------------------------------------|
| 6411 | 040336 | 113760 | 001226 | 000010 | MOVB      | PORTB,\$PCS2(\$R0)  | : SELECT PORT B                             |
| 6412 | 040344 | 013737 | 001226 | 001234 | MOV       | PORTB,\$PTNBR       | : MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 6413 | 040352 | 005037 | 001244 |        | CLR       | CKERR               | : CLEAR THE 'CHECK ERROR' INDICATOR         |
| 6414 | 040356 | 016037 | 000012 | 001126 | MOV       | RPDS1(\$R0),\$SDDAT | : GET CONTENTS OF RPDS1                     |
| 6415 | 040364 | 012737 | 000012 | 001122 | MOV       | #RPDS1,\$SDDADR     | : FORM REGISTER ADDRESS OF ERROR MESSAGE    |
| 6416 | 040372 | 060037 | 001122 |        | ADD       | \$R0,\$SDDADR       | : ADD R#11 BASE ADDRESS                     |
| 6417 | 040376 | 005037 | 001124 |        | CLR       | \$GDDAT             | : WHAT REGISTER SHOULD BE                   |
| 6418 | 040402 | 013737 | 001126 | 001164 | MOV       | \$SDDAT,\$STMP0     | : MOVE REGISTER CONTENTS TO 'STMP0'         |
| 6419 | 040410 | 042737 | 077777 | 001164 | BIC       | #1CATA,\$STMP0      | : SAVE SPECIFIED BITS                       |
| 6420 | 040416 | 023737 | 001124 | 001164 | CMP       | \$GDDAT,\$STMP0     | : COMPARE THE BITS                          |
| 6421 | 040424 | 001414 |        |        | BEQ       | 77\$                | : BR IF OK                                  |
| 6422 | 040426 | 013737 | 001126 | 001174 | MOV       | \$SDDAT,\$STMP4     | : COPY 'BAD DATA'                           |
| 6423 | 040434 | 042737 | 100000 | 001174 | BIC       | #ATA,\$STMP4        | : CLEAR THE MASKED BITS                     |
| 6424 | 040442 | 053737 | 001174 | 001124 | BIS       | \$STMP4,\$GDDAT     | : 'OR' WITH GOOD DATA FOR TYPEOUT           |
| 6425 | 040450 | 104052 |        |        | ERROR     | 52                  | : TYPE MESSAGE 52                           |
| 6426 | 040452 | 005137 | 001244 |        | COM       | CKERR               | : SET THE REGISTER COMPARE ERROR INDICATOR  |
| 6427 | 040456 | 000240 |        |        | 77\$: NOP |                     |                                             |

: CLEAR ATTENTION BIT FOR PORT A

|      |        |        |        |        |            |                    |                     |
|------|--------|--------|--------|--------|------------|--------------------|---------------------|
| 6431 | 040460 | 113760 | 001224 | 000010 | MOVB       | PORTA,\$PCS2(\$R0) | : SELECT PORT #A    |
| 6432 | 040466 | 005060 | 000012 |        | CLR        | RPDS1(\$R0)        | : SEIZE THE DRIVE   |
| 6433 | 040472 | 012760 | 000011 | 000000 | MOV        | #11,\$PCS1(\$R0)   | : ISSUE DRIVE CLEAR |
| 6434 | 040500 | 012760 | 000013 | 000000 | MOV        | #13,\$PCS1(\$R0)   | : RELEASE THE DRIVE |
| 6435 | 040506 | 000004 |        |        | 3\$: SCOPE |                    | : LOOP ?            |

\*\*\*\*\*  
: TEST 32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE  
: \*  
: \* VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.  
: \*  
: \* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.  
: \*  
: \* B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.  
: \*  
: \* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO  
: \* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR  
: \* PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.  
: \*  
: \*\*\*\*\*

|      |        |        |        |  |            |        |                                  |
|------|--------|--------|--------|--|------------|--------|----------------------------------|
| 6451 | 040510 |        |        |  | †ST32: TST | KYBCTL | : PERFORMING ONLY SINGLE TESTS ? |
| 6452 | 040510 | 005737 | 001274 |  |            |        |                                  |



C10

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 119  
CZRJEB.P11 04-NOV-77 13:27 T32

TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE

SEQ 0119

6453 040514 001406  
6454 040516 100002  
6455 040520 000137 002602  
6456 040524 012737 177777 001274 1S:  
6457 040532 112737 000032 001102 2S:  
6458 040540 012737 040562 001106  
6459 040546 012737 040562 001110  
6460 040554 012737 000004 001176  
6461 040562 012706 001100  
6462  
6463  
6464  
6465 040566 113760 001224 000010  
6466 040574 005060 000012  
6467 040600 012760 000011 000000  
6468 040606 012760 000013 000000  
6469 040614 113760 001226 000010  
6470 040622 005060 000012  
6471 040626 012760 000011 000000  
6472 040634 012760 000013 000000  
6473  
6474  
6475  
6476  
6477 040642 113760 001226 000010  
6478 040650 013737 001226 001236  
6479 040656 005060 000012  
6480 040662 013737 001224 001240  
6481  
6482  
6483  
6484  
6485 040670 012760 177777 000014  
6486  
6487  
6488  
6489  
6490 040676 005037 001252  
6491 040702 012737 003720 001254  
6492 040710 113760 001224 000010  
6493 040716 013737 001224 001234  
6494  
6495  
6496  
6497  
6498 040724 005760 000012 1S:  
6499 040730 001004  
6500 040732 005737 001254  
6501 040736 001372  
6502 040740 104036  
6503 040742  
6504 040742 113760 001226 000010 2S:  
6505 040750 013737 001226 001234  
6506  
6507  
6508

```
BEQ 2S ;BR IF NOT
BPL 1S ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR
MOVB #32,STSTNM ;TEST NUMBER
MOV #TEST32,SLPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST32,SLPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,STIMES ;DO 4 ITERATIONS
TEST32: MOV #STACK,SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
CLR RPDS1(RO) ;SEIZE THE DRIVE
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
;*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
;*****
;FORCE AN ERROR
MOV #1,RPERR(RO) ;SET ERROR BITS
;*****
;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #2000,WATCH ;SET WATCH TO 2000 MS
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
;*****
;WAIT FOR DRIVE TO TIMEOUT
1S: TST RPDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
BNE 2S ;BR IF DRIVE RELEASED
TST WATCH ;WATCH AT ZERO ?
BNE 1S ;BR IF NOT
ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
2S: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
;*****
;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET
```

```

6509
6510 040756 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6511 040762 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
6512 040770 012737 000012 001122 MOV @RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6513 040776 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
6514 041002 012737 040000 001124 MOV @EARR, $GDDAT ;WHAT REGISTER SHOULD BE
6515 041010 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6516 041016 042737 137777 001164 BIC @1C4000, $TMP0 ;SAVE SPECIFIED BITS
6517 041024 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6518 041032 001414 BEQ 66$;BR IF OK
6519 041034 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6520 041042 042737 040000 001174 BIC @4000, $TMP4 ;CLEAR THE MASKED BITS
6521 041050 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6522 041056 104023 ERROR 23 ;TYPE MESSAGE 23
6523 041060 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6524 041064 000240 66$: NOP

;:*****
;:THE ERROR REGISTER SHOULD CONTAIN 1'S
6528
6529 041066 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6530 041072 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
6531 041100 012737 000014 001122 MOV @RPER1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6532 041106 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
6533 041112 012737 177777 001124 MOV @177777, $GDDAT ;WHAT REGISTER SHOULD BE
6534 041120 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
6535 041126 001403 BEQ 68$;BR IF OK
6536 041130 104010 ERROR 10 ;REPORT THE ERROR
6537 041132 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6538 041136 000240 68$: NOP

;:*****
;:THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
6542
6543 041140 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6544 041144 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
6545 041152 012737 000012 001122 MOV @RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6546 041160 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
6547 041164 012737 100000 001124 MOV @ATA, $GDDAT ;WHAT REGISTER SHOULD BE
6548 041172 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6549 041200 042737 077777 001164 BIC @1CATA, $TMP0 ;SAVE SPECIFIED BITS
6550 041206 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
6551 041214 001414 BEQ 70$;BR IF OK
6552 041216 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
6553 041224 042737 100000 001174 BIC @ATA, $TMP4 ;CLEAR THE MASKED BITS
6554 041232 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6555 041240 104041 ERROR 41 ;TYPE MESSAGE 41
6556 041242 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6557 041246 000240 70$: NOP

;:*****
;:VERIFY THAT THE DRIVE IS IN NEUTRAL
6563 041250 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR

```

# E10

CZRJEBO, DL CTRLR LGC MACY11 30(1046)  
 CZRJEBO.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 121  
 T32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE

SEQ 0121

|      |        |        |        |        |       |       |                   |                                                    |
|------|--------|--------|--------|--------|-------|-------|-------------------|----------------------------------------------------|
| 6565 | 041254 | 012737 | 000012 | 001122 |       | MOV   | #RPDS1,\$BDAOR    | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT             |
| 6566 | 041262 | 060037 | 001122 |        |       | ADD   | RO,\$BDAOR        | ;ADD THE I/O BASE ADDRESS                          |
| 6567 | 041266 | 012737 | 051700 | 001124 |       | MOV   | #51700,\$GDDAT    | ;COMPARISON CONSTANT                               |
| 6568 | 041274 | 113760 | 001224 | 000010 |       | MOV   | PORTA,RPCS2(RO)   | ;SELECT PORT A.                                    |
| 6569 | 041302 | 016037 | 000012 | 001170 |       | MOV   | RPDS1(RO),\$TMP2  | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 6570 | 041310 | 013737 | 001170 | 001164 |       | MOV   | \$TMP2,\$TMP0     | ;COPY IT INTO '\$TMP0'                             |
| 6571 | 041316 | 042737 | 100100 | 001164 |       | BIC   | #ATA!VV,\$TMP0    | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6572 | 041324 | 113760 | 001226 | 000010 |       | MOV   | PORTB,RPCS2(RO)   | ;SELECT PORT B.                                    |
| 6573 | 041332 | 016037 | 000012 | 001172 |       | MOV   | RPDS1(RO),\$TMP3  | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 6574 | 041340 | 013737 | 001172 | 001166 |       | MOV   | \$TMP3,\$TMP1     | ;COPY IT INTO '\$TMP1'                             |
| 6575 | 041346 | 042737 | 100100 | 001166 |       | BIC   | #ATA!VV,\$TMP1    | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6576 | 041354 | 023737 | 001164 | 001166 |       | CMP   | \$TMP0,\$TMP1     | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 6577 | 041362 | 001006 |        |        |       | BNE   | 72\$              | ;BR IF NOT                                         |
| 6578 | 041364 | 005737 | 001164 |        |       | TST   | \$TMP0            | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 6579 | 041370 | 001045 |        |        |       | BNE   | 74\$              | ;BR IF NOT                                         |
| 6580 | 041372 | 104046 |        |        |       | ERROR | 46                | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 6581 | 041374 | 000137 | 041574 |        |       | JMP   | 76\$              | ;BYPASS THE REST OF THE CHECKS                     |
| 6582 | 041400 | 013737 | 001170 | 001126 | 72\$: | MOV   | \$TMP2,\$BDDAT    | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 6583 | 041406 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR       | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 6584 | 041414 | 113760 | 001226 | 000010 |       | MOV   | PORTB,RPCS2(RO)   | ;SELECT PORT B.                                    |
| 6585 | 041422 | 005737 | 001164 |        |       | TST   | \$TMP0            | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 6586 | 041426 | 001414 |        |        |       | BEQ   | 73\$              | ;BR IF ZERO                                        |
| 6587 | 041430 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR       | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 6588 | 041436 | 013737 | 001172 | 001126 |       | MOV   | \$TMP3,\$BDDAT    | ; 'BAD DATA' FOR ERROR TYPE OUT                    |
| 6589 | 041444 | 113760 | 001224 | 000010 |       | MOV   | PORTA,RPCS2(RO)   | ;SELECT PORT A.                                    |
| 6590 | 041452 | 005737 | 001166 |        |       | TST   | \$TMP1            | ;SEE IF STATUS EQ ZERO FROM PORT B.                |
| 6591 | 041456 | 001012 |        |        |       | BNE   | 74\$              | ;BR IF NOT                                         |
| 6592 | 041460 | 012737 | 177777 | 001250 | 73\$: | MOV   | #-1,RELERR        | ;SET 'RELEASE ERROR' INDICATOR                     |
| 6593 | 041466 | 012760 | 000011 | 000000 |       | MOV   | #11,RPCS1(RO)     | ;CLEAR THE DRIVE                                   |
| 6594 | 041474 | 012760 | 000013 | 000000 |       | MOV   | #13,RPCS1(RO)     | ;RELEASE THE DRIVE                                 |
| 6595 | 041502 | 104026 |        |        |       | ERROR | 26                | ;TYPE ERROR MESSAGE 26                             |
| 6596 | 041504 | 013737 | 001170 | 001126 | 74\$: | MOV   | \$TMP2,\$BDDAT    | ;LOOK FOR BIT FAILURES WHEN RPDS1 READ             |
| 6597 | 041512 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR       | ;CHANGE PORT NUMBER                                |
| 6598 | 041520 | 042737 | 100000 | 001170 |       | BIC   | #ATA,\$TMP2       | ;DON'T CHECK THE ATTN BIT                          |
| 6599 | 041526 | 023737 | 001124 | 001170 |       | CMP   | \$GDDAT,\$TMP2    | ;ALL BITS OK ?                                     |
| 6600 | 041534 | 001401 |        |        |       | BEQ   | 75\$              | ;BR IF OK FROM PORT A.                             |
| 6601 | 041536 | 104007 |        |        |       | ERROR | 7                 | ;REPORT ERROR                                      |
| 6602 | 041540 | 013737 | 001172 | 001126 | 75\$: | MOV   | \$TMP3,\$BDDAT    | ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.       |
| 6603 | 041546 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR       | ;CHANGE PORT NUMBER                                |
| 6604 | 041554 | 042737 | 100000 | 001172 |       | BIC   | #ATA,\$TMP3       | ;DON'T CHECK THE ATTN BIT                          |
| 6605 | 041562 | 023737 | 001124 | 001172 |       | CMP   | \$GDDAT,\$TMP3    | ;SEE IF READ OK FROM PORT B.                       |
| 6606 | 041570 | 001401 |        |        |       | BEQ   | 76\$              | ;BR IF OK                                          |
| 6607 | 041572 | 104007 |        |        |       | ERROR | 7                 | ;REPORT ERROR                                      |
| 6608 | 041574 | 000240 |        |        | 76\$: | NOP   |                   |                                                    |
| 6609 |        |        |        |        |       |       |                   |                                                    |
| 6610 |        |        |        |        |       |       |                   |                                                    |
| 6611 |        |        |        |        |       |       |                   |                                                    |
| 6612 |        |        |        |        |       |       |                   |                                                    |
| 6613 | 041576 | 113760 | 001224 | 000010 |       | MOV   | PORTA,RPCS2(RO)   | ;SELECT PORT A                                     |
| 6614 | 041604 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR       | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 6615 | 041612 | 005037 | 001244 |        |       | CLR   | CKERR             | ;CLEAR THE 'CHECK ERROR' INDICATOR                 |
| 6616 | 041616 | 016037 | 000012 | 001126 |       | MOV   | RPDS1(RO),\$BDDAT | ;GET CONTENTS OF RPDS1                             |
| 6617 | 041624 | 012737 | 000012 | 031122 |       | MOV   | #RPDS1,\$BDAOR    | ;FORM REGISTER ADDRESS OF ERROR MESSAGE            |
| 6618 | 041632 | 060037 | 001122 |        |       | ADD   | RO,\$BDAOR        | ;ADD RH11 BASE ADDRESS                             |
| 6619 | 041636 | 005037 | 001124 |        |       | CLR   | \$GDDAT           | ;WHAT REGISTER SHOULD BE                           |
| 6620 | 041642 | 013737 | 001126 | 001164 |       | MOV   | \$BDDAT,\$TMP0    | ;MOVE REGISTER CONTENTS TO '\$TMP0'                |

\*\*\*\*\*  
 ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

```

6621 041650 042737 077777 001164 BIC #1CATA,$TMPD ;SAVE SPECIFIED BITS
6622 041656 023737 001124 001164 CMP $GDDAT,$TMPD ;COMPARE THE BITS
6623 041664 001414 BEQ 77$;BR IF OK
6624 041666 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6625 041674 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
6626 041702 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6627 041710 104052 ERROR 52 ;TYPE MESSAGE 52
6628 041712 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6629 041716 000240 77$: NOP
6630
6631 ;CLEAR ATTENTION BIT FOR PORT B
6632
6633 041720 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
6634 041726 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
6635 041732 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6636 041740 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6637 041746 000004 3$: SCOPE ;LOOP ?
6638
6639
6640 ;*****
6641 ;*TEST 33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
6642 ;*
6643 ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
6644 ;* BITS ARE SET IN THE DRIVE.
6645 ;*
6646 ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
6647 ;*
6648 ;* B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
6649 ;*
6650 ;* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO'
6651 ;* BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
6652 ;* THAT RPER1 HAS NOT BEEN CLEARED.
6653 ;*
6654 ;* D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
6655 ;*
6656 ;* E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
6657 ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
6658 ;*
6659 ;*****
6660 041750 †ST33:
6661 041750 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6662 041754 001406 BEQ 2$;BR IF NOT
6663 041756 100002 BPL 1$;BR IF JUST ENTERED TEST
6664 041760 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6665 041764 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6666 041772 112737 000033 001102 2$: MOVB #33,$STNM ;TEST NUMBER
6667 042000 012737 042022 001106 MOV #TEST33,$LPADR ;LOAD LOOP ON TEST ADDRESS
6668 042006 012737 042022 001110 MOV #TEST33,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6669 042014 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
6670 042022 012706 001100 TEST33: MOV #STACK,SP ;LOAD THE STACK POINTER
6671
6672 ;CLEAR ATTENTION BITS FOR BOTH PORTS
6673
6674 042026 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
6675 042034 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
6676 042040 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR

```

G10

CZRJEBO DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 123  
CZRJEB.P11 04-NOV-77 13:27

T33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

SEQ 0123

```

6677 042046 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6678 042054 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT #B
6679 042062 005060 000012 000000 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
6680 042066 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6681 042074 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6682 ;:*****
6683 ;SEIZE THE DRIVE THROUGH PORT A
6684
6685
6686 042102 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
6687 042110 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
6688 042116 005060 000012 000000 CLR RPDS1(RO) ;WRITE RPDS1
6689 042122 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
6690
6691 ;:*****
6692 ;FORCE AN ERROR
6693
6694 042130 012760 177777 000014 MOV #-1,RPER1(RO) ;SET ERROR BITS
6695 042136 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
6696 042144 005037 001244 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6697 042150 016037 000000 001126 MOV RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
6698 042156 012737 000000 001122 MOV #RPCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6699 042164 060037 001122 000000 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
6700 042170 012737 004012 001124 MOV #4012,$GDDAT ;WHAT REGISTER SHOULD BE
6701 042176 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6702 042204 042737 173765 001164 BIC #1C4012,$TMP0 ;SAVE SPECIFIED BITS
6703 042212 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6704 042220 001414 000000 000000 BEQ 66$;BR IF OK
6705 042222 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
6706 042230 042737 004012 001174 BIC #4012,$TMP4 ;CLEAR THE MASKED BITS
6707 042236 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6708 042244 104025 000000 000000 ERROR 25 ;TYPE MESSAGE 25
6709 042246 005137 001244 000000 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6710 042252 000240 000000 000000 NOP
6711 042254 005737 001244 000000 TST CKERR ;DID 'GO' BIT RESET ?
6712 042260 001002 000000 000000 BNE +6 ;BR IF NOT
6713 042262 000137 042322 000000 JMP 1$;'GO' BIT RESET
6714 042266 012760 000040 000010 MOV #CLR,RPCS2(RO) ;INIT THE RH11
6715 042274 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
6716 042302 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6717 042310 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE THROUGH PORT A
6718 042316 000137 043036 000000 JMP 2$;BYPASS THE REST OF THE TEST
6719
6720 ;:*****
6721 ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
6722
6723 042322
6724 042322 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B
6725 042330 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6726 042336 005037 001244 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6727 042342 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
6728 042350 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6729 042356 060037 001122 000000 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
6730 042362 005037 001124 000000 CLR $GDDAT ;WHAT REGISTER SHOULD BE
6731 042366 023737 001124 001126 CMP $GDDAT,$SBDDAT ;IS THE REGISTER OK ?
6732 042374 001403 000000 000000 BEQ 68$;BR IF OK

```

|      |        |        |        |        |       |                              |                 |                                                    |
|------|--------|--------|--------|--------|-------|------------------------------|-----------------|----------------------------------------------------|
| 6733 | 042376 | 104024 |        |        | ERROR | 24                           |                 | ;TYPE MESSAGE 24                                   |
| 6734 | 042400 | 005137 | 001244 |        | COM   | CKERR                        |                 | ;SET THE REGISTER COMPARE ERROR INDICATOR          |
| 6735 | 042404 | 000240 |        |        | 68\$: | NOP                          |                 |                                                    |
| 6736 | 042406 | 113760 | 001224 | 000010 | MOV   | PORTA,RPCS2(RO)              |                 | ;SELECT PORT A                                     |
| 6737 | 042414 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                  |                 | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 6738 | 042422 | 005037 | 001244 |        | CLR   | CKERR                        |                 | ;CLEAR THE 'CHECK ERROR' INDICATOR                 |
| 6739 | 042426 | 016037 | 000014 | 001126 | MOV   | RPER1(RO), \$BDDAT           |                 | ;GET CONTENTS OF RPER1                             |
| 6740 | 042434 | 012737 | 000014 | 001122 | MOV   | #RPER1, \$BDDADR             |                 | ;FORM REGISTER ADDRESS OF ERROR MESSAGE            |
| 6741 | 042442 | 060037 | 001122 |        | ADD   | RO, \$BDDADR                 |                 | ;ADD RH11 BASE ADDRESS                             |
| 6742 | 042446 | 012737 | 177777 | 001124 | MOV   | #177777, \$GDDAT             |                 | ;WHAT REGISTER SHOULD BE                           |
| 6743 | 042454 | 023737 | 001124 | 001126 | CMP   | \$GDDAT, \$BDDAT             |                 | ;IS THE REGISTER OK ?                              |
| 6744 | 042462 | 001403 |        |        | BEG   | 70\$                         |                 | ;BR IF OK                                          |
| 6745 | 042464 | 104010 |        |        | ERROR | 10                           |                 | ;REPORT THE ERROR                                  |
| 6746 | 042466 | 005137 | 001244 |        | COM   | CKERR                        |                 | ;SET THE REGISTER COMPARE ERROR INDICATOR          |
| 6747 | 042472 | 000240 |        |        | 70\$: | NOP                          |                 |                                                    |
| 6748 |        |        |        |        |       |                              |                 |                                                    |
| 6749 |        |        |        |        |       |                              |                 |                                                    |
| 6750 |        |        |        |        |       |                              |                 | ;*****                                             |
| 6751 |        |        |        |        |       |                              |                 | ;CLEAR THE ERRORS THROUGH PORT A                   |
| 6752 | 042474 | 012760 | 000011 | 000000 | MOV   | #11,RPCS1(RO)                |                 | ;ISSUE A DRIVE CLEAR                               |
| 6753 |        |        |        |        |       |                              |                 |                                                    |
| 6754 |        |        |        |        |       |                              |                 | ;*****                                             |
| 6755 |        |        |        |        |       |                              |                 |                                                    |
| 6756 |        |        |        |        |       |                              |                 | ;RELEASE THE DRIVE FROM PORT A                     |
| 6757 |        |        |        |        |       |                              |                 |                                                    |
| 6758 | 042502 | 113760 | 001224 | 000010 | MOV   | PORTA,RPCS2(RO)              |                 | ;SELECT PORT A                                     |
| 6759 | 042510 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                  |                 | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 6760 | 042516 | 012760 | 000013 | 000000 | MOV   | #13,RPCS1(RO)                |                 | ;ISSUE RELEASE THROUGH PORT A                      |
| 6761 |        |        |        |        |       |                              |                 |                                                    |
| 6762 |        |        |        |        |       |                              |                 | ;VERIFY THAT THE DRIVE IS IN NEUTRAL               |
| 6763 |        |        |        |        |       |                              |                 |                                                    |
| 6764 | 042524 | 005037 | 001250 |        | CLR   | RELEARR                      |                 | ;CLEAR THE 'RELEASE ERROR' INDICATOR               |
| 6765 | 042530 | 012737 | 000012 | 001122 | MOV   | #RPDS1, \$BDDADR             |                 | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT             |
| 6766 | 042536 | 060037 | 001122 |        | ADD   | RO, \$BDDADR                 |                 | ;ADD THE I/O BASE ADDRESS                          |
| 6767 | 042542 | 012737 | 011700 | 001124 | MOV   | #MOL!PGM!DPR!DRY!VV, \$GDDAT |                 | ;COMPARISON CONSTANT                               |
| 6768 | 042550 | 113760 | 001224 | 000010 | MOV   | PORTA,RPCS2(RO)              |                 | ;SELECT PORT A.                                    |
| 6769 | 042556 | 016037 | 000012 | 001170 | MOV   | RPDS1(RO), \$TMP2            |                 | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 6770 | 042564 | 013737 | 001170 | 001164 | MOV   | \$TMP2, \$TMP0               |                 | ;COPY IT INTO 'TMP0'                               |
| 6771 | 042572 | 042737 | 100100 | 001164 | BIC   | #ATA!VV, \$TMP0              |                 | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6772 | 042600 | 113760 | 001226 | 000010 | MOV   | PORTB,RPCS2(RO)              |                 | ;SELECT PORT B.                                    |
| 6773 | 042606 | 016037 | 000012 | 001172 | MOV   | RPDS1(RO), \$TMP3            |                 | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 6774 | 042614 | 013737 | 001172 | 001166 | MOV   | \$TMP3, \$TMP1               |                 | ;COPY IT INTO 'TMP1'                               |
| 6775 | 042622 | 042737 | 100100 | 001166 | BIC   | #ATA!VV, \$TMP1              |                 | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6776 | 042630 | 023737 | 001164 | 001166 | CMP   | \$TMP0, \$TMP1               |                 | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 6777 | 042636 | 001006 |        |        | BNE   | 72\$                         |                 | ;BR IF NOT                                         |
| 6778 | 042640 | 005737 | 001164 |        | TST   | \$TMP0                       |                 | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 6779 | 042644 | 001045 |        |        | BNE   | 74\$                         |                 | ;BR IF NOT                                         |
| 6780 | 042646 | 104046 |        |        | ERROR | 46                           |                 | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 6781 | 042650 | 000137 | 043034 |        | JMP   | 76\$                         |                 | ;BYPASS THE REST OF THE CHECKS                     |
| 6782 | 042654 | 013737 | 001170 | 001126 | 72\$: | MOV                          | \$TMP2, \$BDDAT | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 6783 | 042662 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                  |                 | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 6784 | 042670 | 113760 | 001226 | 000010 | MOV   | PORTB,RPCS2(RO)              |                 | ;SELECT PORT B.                                    |
| 6785 | 042676 | 005737 | 001164 |        | MOV   | \$TMP0                       |                 | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 6786 | 042702 | 001414 |        |        | BEG   | 73\$                         |                 | ;BR IF ZERO                                        |
| 6787 | 042704 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                  |                 | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 6788 | 042712 | 013737 | 001172 | 001126 | MOV   | \$TMP3, \$BDDAT              |                 | ; 'BAD DATA' FOR ERROR TYPE OUT                    |

```

6789 042720 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
6790 042726 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6791 042732 001012 74$: BNE 74$;BR IF NOT
6792 042734 012737 177777 001250 MOV #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
6793 042742 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
6794 042750 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6795 042756 104026 001170 001126 ERROR 26 ;TYPE ERROR MESSAGE 26
6796 042760 013737 001224 001234 MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6797 042766 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6798 042774 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6799 043002 001401 75$: BEQ 75$;BR IF OK FROM PORT A.
6800 043004 104007 75$: ERROR 7 ;REPORT ERROR
6801 043006 013737 001172 001126 MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6802 043014 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6803 043022 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6804 043030 001401 76$: BEQ 76$;BR IF OK
6805 043032 104007 76$: ERROR 7 ;REPORT ERROR
6806 043034 000240 2$: NOP
6807 043036 000004 2$: SCOPE ;LOOP ?

```

```

*TEST 34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
*
*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
* BITS ARE SET IN THE DRIVE.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
* B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
*
* C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO'
* BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
* THAT RPER1 HAS NOT BEEN CLEARED.
*
* D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
*
* E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*

```

```

6829 043040 ST34:
6830 043040 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6831 043044 001406 2$: BEQ 2$;BR IF NOT
6832 043046 100002 1$: BPL 1$;BR IF JUST ENTERED TEST
6833 043050 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
6834 043054 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
6835 043062 112737 000034 001102 2$: MOVB #34,$STNM ;TEST NUMBER
6836 043070 012737 043112 001106 MOV #TEST34,$LPADR ;LOAD LOOP ON TEST ADDRESS
6837 043076 012737 043112 001110 MOV #TEST34,$LPERR ;LOAD LOOP ON ERROR ADDRESS
6838 043104 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
6839 043112 012706 001100 TEST34: MOV #STACK,$SP ;LOAD THE STACK POINTER
6840
6841
6842
6843 043116 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
6844 043124 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE

```

```

6845 043130 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6846 043136 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6847 043144 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT #B
6848 043152 005060 000012 000000 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
6849 043156 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
6850 043164 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
6851 ;*****
6852 ;SEIZE THE DRIVE THROUGH PORT B
6853
6854
6855 043172 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B
6856 043200 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
6857 043206 005060 000012 000000 CLR RPDS1(RO) ;WRITE RPDS1
6858 043212 013737 001224 001240 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
6859
6860 ;*****
6861 ;FORCE AN ERROR
6862
6863 043220 012760 177777 000014 MOV #-1,RPER1(RO) ;SET ERROR BITS
6864 043226 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
6865 043234 005037 001244 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6866 043240 016037 000000 001126 MOV RPCS1(RO),SBDDAT ;GET CONTENTS OF RPCS1
6867 043246 012737 000000 001122 MOV #RPCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6868 043254 060037 001122 000000 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
6869 043260 012737 004012 001124 MOV #4012,$GDDAT ;WHAT REGISTER SHOULD BE
6870 043266 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
6871 043274 042737 173765 001164 BIC #1C4012,$TMP0 ;SAVE SPECIFIED BITS
6872 043302 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
6873 043310 001414 000000 000000 BEQ 66$;BR IF OK
6874 043312 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
6875 043320 042737 004012 001174 BIC #4012,$TMP4 ;CLEAR THE MASKED BITS
6876 043326 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
6877 043334 104025 000000 000000 ERROR 25 ;TYPE MESSAGE 25
6878 043336 005137 001244 000000 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
6879 043342 000240 000000 000000 NOP
6880 043344 005737 001244 000000 TST CKERR ;DID 'GO' BIT RESET ?
6881 043350 001002 000000 000000 BNE .+6 ;BR IF NOT
6882 043352 000137 043412 000000 JMP 1$;'GO' BIT RESET
6883 043356 012760 000040 000010 MOV #CLR,RPCS2(RO) ;INIT THE RH11
6884 043364 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B
6885 043372 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6886 043400 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE THROUGH PORT B
6887 043406 000137 044126 000000 JMP 2$;BYPASS THE REST OF THE TEST
6888
6889 ;*****
6890 ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
6891
6892 1$:
6893 043412 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
6894 043420 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6895 043426 005037 001244 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
6896 043432 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
6897 043440 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6898 043446 060037 001122 000000 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
6899 043452 005037 001124 000000 CLR $GDDAT ;WHAT REGISTER SHOULD BE
6900 043456 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?

```



K10

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 127  
CZRJEB.P11 04-NOV-77 13:27

T34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET

SEQ 0127

|      |        |        |        |        |       |                                  |       |                                                    |
|------|--------|--------|--------|--------|-------|----------------------------------|-------|----------------------------------------------------|
| 6901 | 043464 | 001403 |        |        | BEQ   | 68\$                             |       | ;BR IF OK                                          |
| 6902 | 043466 | 104024 |        |        | ERROR | 24                               |       | ;TYPE MESSAGE 24                                   |
| 6903 | 043470 | 005137 | 001244 |        | COM   | CKERR                            |       | ;SET THE REGISTER COMPARE ERROR INDICATOR          |
| 6904 | 043474 | 000240 |        |        | NOP   |                                  | 68\$: |                                                    |
| 6905 | 043476 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                  |       | ;SELECT PORT B                                     |
| 6906 | 043504 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                      |       | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 6907 | 043512 | 005037 | 001244 |        | CLR   | CKERR                            |       | ;CLEAR THE 'CHECK ERROR' INDICATOR                 |
| 6908 | 043516 | 016037 | 000014 | 001126 | MOV   | RPER1(RO), \$BDDAT               |       | ;GET CONTENTS OF RPER1                             |
| 6909 | 043524 | 012737 | 000014 | 001122 | MOV   | #RPER1, \$BDADR                  |       | ;FORM REGISTER ADDRESS OF ERROR MESSAGE            |
| 6910 | 043532 | 060037 | 001122 |        | ADD   | RO, \$BDADR                      |       | ;ADD RH11 BASE ADDRESS                             |
| 6911 | 043536 | 012737 | 177777 | 001124 | MOV   | #177777, \$GDDAT                 |       | ;WHAT REGISTER SHOULD BE                           |
| 6912 | 043544 | 023737 | 001124 | 001126 | CMP   | \$GDDAT, \$BDDAT                 |       | ;IS THE REGISTER OK ?                              |
| 6913 | 043552 | 001403 |        |        | BEQ   | 70\$                             |       | ;BR IF OK                                          |
| 6914 | 043554 | 104010 |        |        | ERROR | 10                               |       | ;REPORT THE ERROR                                  |
| 6915 | 043556 | 005137 | 001244 |        | COM   | CKERR                            |       | ;SET THE REGISTER COMPARE ERROR INDICATOR          |
| 6916 | 043562 | 000240 |        |        | NOP   |                                  | 70\$: |                                                    |
| 6917 |        |        |        |        |       |                                  |       |                                                    |
| 6918 |        |        |        |        |       |                                  |       | ::*****                                            |
| 6919 |        |        |        |        |       |                                  |       | ;CLEAR THE ERRORS THROUGH PORT B                   |
| 6920 |        |        |        |        |       |                                  |       |                                                    |
| 6921 | 043564 | 012760 | 000011 | 000000 | MOV   | #11,RPCS1(RO)                    |       | ;ISSUE A DRIVE CLEAR                               |
| 6922 |        |        |        |        |       |                                  |       |                                                    |
| 6923 |        |        |        |        |       |                                  |       | ::*****                                            |
| 6924 |        |        |        |        |       |                                  |       |                                                    |
| 6925 |        |        |        |        |       |                                  |       | ;RELEASE THE DRIVE FROM PORT B                     |
| 6926 |        |        |        |        |       |                                  |       |                                                    |
| 6927 | 043572 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                  |       | ;SELECT PORT B                                     |
| 6928 | 043600 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                      |       | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 6929 | 043606 | 012760 | 000013 | 000000 | MOV   | #13,RPCS1(RO)                    |       | ;ISSUE RELEASE THROUGH PORT B                      |
| 6930 |        |        |        |        |       |                                  |       |                                                    |
| 6931 |        |        |        |        |       |                                  |       | ;VERIFY THAT THE DRIVE IS IN NEUTRAL               |
| 6932 |        |        |        |        |       |                                  |       |                                                    |
| 6933 | 043614 | 005037 | 001250 |        | CLR   | RELERR                           |       | ;CLEAR THE 'RELEASE ERROR' INDICATOR               |
| 6934 | 043620 | 012737 | 000012 | 001122 | MOV   | #RPDS1, \$BDADR                  |       | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT             |
| 6935 | 043626 | 060037 | 001122 |        | ADD   | RO, \$BDADR                      |       | ;ADD THE I/O BASE ADDRESS                          |
| 6936 | 043632 | 012737 | 011700 | 001124 | MOV   | #MOL, PGM, DPR, DRY, VV, \$GDDAT |       | ;COMPARISON CONSTANT                               |
| 6937 | 043640 | 113760 | 001224 | 000010 | MOVB  | PORTA,RPCS2(RO)                  |       | ;SELECT PORT A.                                    |
| 6938 | 043646 | 016037 | 000012 | 001170 | MOV   | RPDS1(RO), \$TMP2                |       | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 6939 | 043654 | 013737 | 001170 | 001164 | MOV   | \$TMP2, \$TMP0                   |       | ;COPY IT INTO 'TMP0'                               |
| 6940 | 043662 | 042737 | 100100 | 001164 | BIC   | #ATA!VV, \$TMP0                  |       | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6941 | 043670 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                  |       | ;SELECT PORT B.                                    |
| 6942 | 043676 | 016037 | 000012 | 001172 | MOV   | RPDS1(RO), \$TMP3                |       | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 6943 | 043704 | 013737 | 001172 | 001166 | MOV   | \$TMP3, \$TMP1                   |       | ;COPY IT INTO 'TMP1'                               |
| 6944 | 043712 | 042737 | 100100 | 001166 | BIC   | #ATA!VV, \$TMP1                  |       | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 6945 | 043720 | 023737 | 001164 | 001166 | CMP   | \$TMP0, \$TMP1                   |       | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 6946 | 043726 | 001006 |        |        | BNE   | 72\$                             |       | ;BR IF NOT                                         |
| 6947 | 043730 | 005737 | 001164 |        | TST   | \$TMP0                           |       | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 6948 | 043734 | 001045 |        |        | BNE   | 74\$                             |       | ;BR IF NOT                                         |
| 6949 | 043736 | 104046 |        |        | ERROR | 46                               |       | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 6950 | 043740 | 000137 | 044124 |        | JMP   | 76\$                             |       | ;BYPASS THE REST OF THE CHECKS                     |
| 6951 | 043744 | 013737 | 001170 | 001126 | MOV   | \$TMP2, \$BDDAT                  |       | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 6952 | 043752 | 013737 | 001226 | 001234 | MOV   | PORTB,PTNBR                      |       | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 6953 | 043760 | 113760 | 001226 | 000010 | MOVB  | PORTB,RPCS2(RO)                  |       | ;SELECT PORT B.                                    |
| 6954 | 043766 | 005737 | 001164 |        | TST   | \$TMP0                           |       | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 6955 | 043772 | 001414 |        |        | BEQ   | 73\$                             |       | ;BR IF ZERO                                        |
| 6956 | 043774 | 013737 | 001224 | 001234 | MOV   | PORTA,PTNBR                      |       | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |

```

6957 044002 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
6958 044010 113760 001224 000010 MOV PORTA,RPDS2(RO) ;SELECT PORT A.
6959 044016 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
6960 044022 001012 BNE 74$;BR IF NOT
6961 044024 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
6962 044032 012760 000011 000000 MOV #11,RPDS1(RO) ;CLEAR THE DRIVE
6963 044040 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE
6964 044046 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
6965 044050 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6966 044056 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
6967 044064 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
6968 044072 001401 BEQ 75$;BR IF OK FROM PORT A.
6969 044074 104007 ERROR 7 ;REPORT ERROR
6970 044076 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6971 044104 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
6972 044112 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
6973 044120 001401 BEQ 76$;BR IF OK
6974 044122 104007 ERROR 7 ;REPORT ERROR
6975 044124 000240 76$: NOP
6976 044126 000004 2$: SCOPE ;LOOP ?
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012

```

```

*TEST 35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
*
*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.
*
* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
* TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
*
* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
* BIT IS SET.

```

```

6995 044130
6996 044130 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
6997 044134 001406 BEQ 2$;BR IF NOT
6998 044136 100002 BPL 1$;BR IF JUST ENTERED TEST
6999 044140 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
7000 044144 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
7001 044152 112737 000035 001102 2$: MOV #35,$STNM ;TEST NUMBER
7002 044160 012737 044202 001106 MOV #TEST35,$LPADR ;LOAD LOOP ON TEST ADDRESS
7003 044166 012737 044202 001110 MOV #TEST35,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7004 044174 012737 000004 001176 MOV #4,$TIMES ;DO 4 ITERATIONS
7005 044202 012706 001100 TEST35: MOV #STACK,SP ;LOAD THE STACK POINTER
7006
7007
7008 ;CLEAR ATTENTION BITS FOR BOTH PORTS
7009 044206 113760 001224 000010 MOV PORTA,RPDS2(RO) ;SELECT PORT #A
7010 044214 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
7011 044220 012760 000011 000000 MOV #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
7012 044226 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE

```

M10

CZRJEBO, DL CTRLR LGC MACY11 30(1046)  
 CZRJEB.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 129  
 T35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

SEQ 0129

```

7013 044234 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
7014 044242 005060 000012 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
7015 044246 012760 000011 000000 MOV #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
7016 044254 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE
7017
7018 ;*****
7019
7020 ;SEIZE THE DRIVE THROUGH PORT A
7021
7022 044262 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
7023 044270 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
7024 044276 005060 000012 000012 CLR RPDS1(RO) ;WRITE RPDS1
7025 044302 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
7026
7027 ;*****
7028 ;WAIT 500 MS
7029
7030
7031 ;*****
7032 ;START THE TIMER
7033
7034 044310 005037 001252 000000 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
7035 044314 012737 000764 001254 MOV #500.,WATCH ;SET WATCH TO 500 MS
7036 044322 005737 001254 001254 1$: TST WATCH ;WATCH EQUAL TO ZERO
7037 044326 001375 001254 001254 BNE 1$;BR IF NOT
7038
7039 ;*****
7040 ;START THE TIMER
7041
7042 044330 005037 001252 000000 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
7043 044334 012737 003720 001254 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
7044
7045 ;*****
7046 ;RETRIGGER THE TIMEOUT ONE-SHOT
7047
7048 044342 005760 000012 000000 TST RPDS1(RO) ;RETRIGGER THE ONE-SHOT
7049 044346 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
7050 044354 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7051 044362 005760 000012 000012 2$: TST RPDS1(RO) ;WAIT FOR TIMEOUT
7052 044366 001004 000012 000012 BNE 3$;BR IF TIMEOUT OCCURRED
7053 044370 005737 001254 001254 TST WATCH ;WATCH EQUAL TO ZERO ?
7054 044374 001372 001254 001254 BNE 2$;BR IF NOT
7055 044376 104036 001252 001272 ERROR 3$;NO TIMEOUT WITHIN 2 SECONDS
7056 044400 013737 001252 001272 3$: MOV TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
7057
7058 ;*****
7059
7060 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7061
7062 044406 005037 001250 000000 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7063 044412 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7064 044420 060037 001122 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
7065 044424 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
7066 044432 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
7067 044440 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7068 044446 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'

```

|      |        |        |        |        |       |       |                 |                                                    |
|------|--------|--------|--------|--------|-------|-------|-----------------|----------------------------------------------------|
| 7069 | 044454 | 042737 | 100100 | 001164 |       | BIC   | #ATA!VV,STMP0   | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 7070 | 044462 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO) | :SELECT PORT B.                                    |
| 7071 | 044470 | 016037 | 000012 | 001172 |       | MOV   | RPDS1(RO),STMP3 | :GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 7072 | 044476 | 013737 | 001172 | 001166 |       | MOV   | STMP3,STMP1     | :COPY IT INTO 'STMP1'                              |
| 7073 | 044504 | 042737 | 100100 | 001166 |       | BIC   | #ATA!VV,STMP1   | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 7074 | 044512 | 023737 | 001164 | 001166 |       | CMP   | STMP0,STMP1     | :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 7075 | 044520 | 001006 |        |        |       | BNE   | 66\$            | :BR IF NOT                                         |
| 7076 | 044522 | 005737 | 001164 |        |       | TST   | STMP0           | :REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 7077 | 044526 | 001045 |        |        |       | BNE   | 68\$            | :BR IF NOT                                         |
| 7078 | 044530 | 104046 |        |        |       | ERROR | 46              | :REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 7079 | 044532 | 000137 | 044716 |        |       | JMP   | 70\$            | :BYPASS THE REST OF THE CHECKS                     |
| 7080 | 044536 | 013737 | 001170 | 001126 | 66\$: | MOV   | STMP2,\$BDDAT   | :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 7081 | 044544 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR     | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 7082 | 044552 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO) | :SELECT PORT B.                                    |
| 7083 | 044560 | 005737 | 001164 |        |       | TST   | STMP0           | :SEE IF STATUS EQ 0 FROM PORT A.                   |
| 7084 | 044564 | 001414 |        |        |       | BEQ   | 67\$            | :BR IF ZERO                                        |
| 7085 | 044566 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR     | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 7086 | 044574 | 013737 | 001172 | 001126 |       | MOV   | STMP3,\$BDDAT   | : 'BAD DATA' FOR ERROR TYPE OUT                    |
| 7087 | 044602 | 113760 | 001224 | 000010 |       | MOVB  | PORTA,RPCS2(RO) | :SELECT PORT A.                                    |
| 7088 | 044610 | 005737 | 001166 |        |       | TST   | STMP1           | :SEE IF STATUS EQ ZERO FROM PORT B.                |
| 7089 | 044614 | 001012 |        |        |       | BNE   | 68\$            | :BR IF NOT                                         |
| 7090 | 044616 | 012737 | 177777 | 001250 | 67\$: | MOV   | #-1,RELEA       | :SET 'RELEASE ERROR' INDICATOR                     |
| 7091 | 044624 | 012760 | 000011 | 000000 |       | MOV   | #11,RPCS1(RO)   | :CLEAR THE DRIVE                                   |
| 7092 | 044632 | 012760 | 000013 | 000000 |       | MOV   | #13,RPCS1(RO)   | :RELEASE THE DRIVE                                 |
| 7093 | 044640 | 104022 |        |        |       | ERROR | 22              | :TYPE ERROR MESSAGE 22                             |
| 7094 | 044642 | 013737 | 001170 | 001126 | 68\$: | MOV   | STMP2,\$BDDAT   | :LOOK FOR BIT FAILURES WHEN RPDS1 READ             |
| 7095 | 044650 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR     | :CHANGE PORT NUMBER                                |
| 7096 | 044656 | 023737 | 001124 | 001170 |       | CMP   | SGDAT,STMP2     | :ALL BITS OK ?                                     |
| 7097 | 044664 | 001401 |        |        |       | BEQ   | 69\$            | :BR IF OK FROM PORT A.                             |
| 7098 | 044666 | 104007 |        |        |       | ERROR | 7               | :REPORT ERROR                                      |
| 7099 | 044670 | 013737 | 001172 | 001126 | 69\$: | MOV   | STMP3,\$BDDAT   | :CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.       |
| 7100 | 044676 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR     | :CHANGE PORT NUMBER                                |
| 7101 | 044704 | 023737 | 001124 | 001172 |       | CMP   | SGDAT,STMP3     | :SEE IF READ OK FROM PORT B.                       |
| 7102 | 044712 | 001401 |        |        |       | BEQ   | 70\$            | :BR IF OK                                          |
| 7103 | 044714 | 104007 |        |        |       | ERROR | 7               | :REPORT ERROR                                      |
| 7104 | 044716 | 000240 |        |        | 70\$: | NOP   |                 |                                                    |

\*\*\*\*\*  
:CHECK THE TIME FROM RETRIGGER TO TIMEOUT

|      |        |        |        |        |      |       |              |                                              |
|------|--------|--------|--------|--------|------|-------|--------------|----------------------------------------------|
| 7109 | 044720 | 023737 | 001272 | 001260 |      | CMP   | TIMES,TIMEAP | :MEASURED TIME GREATER THAN +25% TOLERANCE ? |
| 7110 | 044726 | 003004 |        |        |      | BGT   | 4\$          | :BR IF GREATER                               |
| 7111 | 044730 | 023737 | 001272 | 001262 |      | CMP   | TIMES,TIMEAM | :MEASURED TIME LESS THAN -25% TOLERANCE      |
| 7112 | 044736 | 002001 |        |        |      | BGE   | +4           | :BR IF NOT                                   |
| 7113 | 044740 | 104025 |        |        | 4\$: | ERROR | 25           | :REPORT THE ERROR                            |
| 7114 | 044742 | 000004 |        |        |      | SCOPE |              | :LOOP ?                                      |

\*\*\*\*\*  
:TEST 36 TEST TIMEOUT RETRIGGER THROUGH PORT 'B'

```

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

```

\* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)  
\* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

7125  
7126  
7127  
7128  
7129  
7130  
7131  
7132 044744  
7133 044744 005737 001274  
7134 044750 001406  
7135 044752 100002  
7136 044754 000137 002602  
7137 044760 012737 177777 001274  
7138 044766 112737 000036 001102  
7139 044774 012737 045016 001106  
7140 045002 012737 045016 001110  
7141 045010 012737 000004 001176  
7142 045016 012706 001100  
7143  
7144  
7145  
7146 045022 113760 001224 000010  
7147 045030 005060 000012  
7148 045034 012760 000011 000000  
7149 045042 012760 000013 000000  
7150 045050 113760 001226 000010  
7151 045056 005060 000012  
7152 045062 012760 000011 000000  
7153 045070 012760 000013 000000  
7154  
7155  
7156  
7157  
7158  
7159 045076 113760 001226 000010  
7160 045104 013737 001226 001236  
7161 045112 005060 000012  
7162 045116 013737 001224 001240  
7163  
7164  
7165  
7166  
7167  
7168  
7169  
7170  
7171 045124 005037 001252  
7172 045130 012737 000764 001254  
7173 045136 005737 001254  
7174 045142 001375  
7175  
7176  
7177  
7178  
7179 045144 005037 001252  
7180 045150 012737 003720 001254

```

↑ST36: 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 #36,$TSTNM ;TEST NUMBER
MOV #TEST36,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST36,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST36: MOV #STACK,$P ;LOAD THE STACK POINTER

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

;*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RPCS2(R0) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(R0) ;WRITE RPDS1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

;*****
;WAIT 500 MS

;*****
;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
1$: MOV #500.,WATCH ;SET WATCH TO 500 MS
TST WATCH ;WATCH EQUAL TO ZERO
BNE 15 ;BR IF NOT

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

```

7181
7182
7183
7184
7185 045156 005760 000012
7186 045162 113760 001224 000010
7187 045170 013737 001224 001234
7188 045176 005760 000012
7189 045202 001004
7190 045204 005737 001254
7191 045210 001372
7192 045212 104036
7193 045214 013737 001252 001272
7194
7195
7196
7197
7198
7199 045222 005037 001250
7200 045226 012737 000012 001122
7201 045234 060037 001122
7202 045240 012737 011700 001124
7203 045246 113760 001224 000010
7204 045254 016037 000012 001170
7205 045262 013737 001170 001164
7206 045270 042737 100100 001164
7207 045276 113760 001226 000010
7208 045304 016037 000012 001172
7209 045312 013737 001172 001166
7210 045320 042737 100100 001166
7211 045326 023737 001164 001166
7212 045334 001006
7213 045336 005737 001164
7214 045342 001045
7215 045344 104046
7216 045346 000137 045532
7217 045352 013737 001170 001126 66$:
7218 045360 013737 001226 001234
7219 045366 113760 001226 000010
7220 045374 005737 001164
7221 045400 001414
7222 045402 013737 001224 001234
7223 045410 013737 001172 001126
7224 045416 113760 001224 000010
7225 045424 005737 001166
7226 045430 001012
7227 045432 012737 177777 001250 67$:
7228 045440 012760 000011 000000
7229 045446 012760 000013 000000
7230 045454 104022
7231 045456 013737 001170 001126 68$:
7232 045464 013737 001224 001234
7233 045472 023737 001124 001170
7234 045500 001401
7235 045502 104007
7236 045504 013737 001172 001126 69$:

```

\*\*\*\*\*  
;RETRIGGER THE TIMEOUT ONE-SHOT

```

TST RPDS1(RO) ;RETRIGGER THE ONE-SHOT
MOV PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2$: TST RPDS1(RO) ;WAIT FOR TIMEOUT
BNE 3$;BR IF TIMEOUT OCCURRED
TST WATCH ;WATCH EQUAL TO ZERO ?
BNE 2$;BR IF NOT
ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
MOV TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
3$:

```

\*\*\*\*\*

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL:PGM:DPR:DRY,VV,$GDDAT ;COMPARISON CONSTANT
MOV PORTA,RPCS2(RO) ;SELECT PORT A.
MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOV PORTB,RPCS2(RO) ;SELECT PORT B.
MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 66$;BR IF NOT
TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 68$;BR IF NOT
ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
JMP 70$;BYPASS THE REST OF THE CHECKS
66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOV PORTB,RPCS2(RO) ;SELECT PORT B.
TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
BEQ 67$;BR IF ZERO
MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
MOV PORTA,RPCS2(RO) ;SELECT PORT A.
TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
BNE 68$;BR IF NOT
MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
ERROR 22 ;TYPE ERROR MESSAGE 22
MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
MOV PORTA,PTNBR ;CHANGE PORT NUMBER
CMP $GDDAT,$TMP2 ;ALL BITS OK ?
BEQ 69$;BR IF OK FROM PORT A.
ERROR 7 ;REPORT ERROR
MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
69$:

```

```

7237 045512 013737 001226 001234
7238 045520 023737 001124 001172
7239 045526 001401
7240 045530 104007
7241 045532 000240
7242
7243
7244
7245
7246 045534 023737 001272 001266
7247 045542 003004
7248 045544 023737 001272 001270
7249 045552 002001
7250 045554 104025
7251 045556 000004
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272 045560
7273 045560 005737 001274
7274 045564 001406
7275 045566 100002
7276 045570 000137 002602
7277 045574 012737 177777 001274
7278 045602 112737 000037 001102
7279 045610 012737 045632 001106
7280 045616 012737 045632 001110
7281 045624 012737 000004 001176
7282 045632 012706 001100
7283
7284
7285
7286 045636 113760 001224 000010
7287 045644 005060 000012
7288 045650 012760 000011 000000
7289 045656 012760 000013 000000
7290 045664 113760 001226 000010
7291 045672 005060 000012
7292 045676 012760 000011 000000

```

```

MOV PORTB,PTNBR ;CHANGE PORT NUMBER
CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
BEQ 70$;BR IF OK
ERROR 7 ;REPORT ERROR
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
ERROR 25 ;REPORT THE ERROR
SCOPE ;LOOP ?

*TEST 37 TEST PORT 'A' ATTENTION AFTER A COMMAND
*
*
*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
*COMMAND.
*
* A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
*
* B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
*'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.
*

*ST37:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$;BR IF NOT
BPL 1$;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
MOVB #37,$STSTNM ;TEST NUMBER
MOV #TEST37,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST37,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST37: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

```

E11

CZRJEBO DL CTRLR LGC MACY11 30(1046)  
CZRJEB.P11 04-NOV-77 13:27

04-NOV-77 17:48 PAGE 134  
T37 TEST PORT 'A' ATTENTION AFTER A COMMAND

SEG 0134

```

7293 045704 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
7294 045712 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
7295 045720 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7296 045726 013737 001224 001236 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
7297
7298 ;:*****
7299 ;DO A RECALIBRATE THROUGH PORT A
7300
7301 045734 012760 000007 000000 MOV #7,RPCS1(RO) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
7302
7303 ;:*****
7304 ;WAIT FOR DRIVE TO FINISH
7305
7306 045742 032760 000200 000012 BIT #DRY,RPDS1(RO) ;WAIT FOR DRIVE TO FINISH
7307 045750 001774 BEQ .-6 ;BR IF NOT FINISHED
7308
7309 ;:*****
7310 ;CONFIRM THAT ATTENTION IS SET FOR PORT A
7311
7312 045752 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7313 045756 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
7314 045764 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7315 045772 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
7316 045776 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
7317 046004 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7318 046012 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
7319 046020 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
7320 046026 001414 BEQ 64$;BR IF OK
7321 046030 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
7322 046036 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
7323 046044 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7324 046052 104032 ERROR 32 ;TYPE MESSAGE 32
7325 046054 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7326 046060 000240
7327 046062 113760 001226 000010 64$: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
7328 046070 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7329
7330 ;:*****
7331 ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
7332
7333 046076 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7334 046102 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
7335 046110 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7336 046116 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
7337 046122 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7338 046126 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
7339 046134 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
7340 046142 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
7341 046150 001414 BEQ 66$;BR IF OK
7342 046152 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
7343 046160 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
7344 046166 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
7345 046174 104032 ERROR 32 ;TYPE MESSAGE 32
7346 046176 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7347 046202 000240
7348 66$: NOP

```



F11

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 135  
CZRJEB.P11 04-NOV-77 13:27

T37 TEST PORT 'A' ATTENTION AFTER A COMMAND

SEQ 0135

```

7349 ;:*****
7350 ;
7351 ;RELEASE THE DRIVE FROM PORT A
7352 ;
7353 046204 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A
7354 046212 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7355 046220 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7356 ;
7357 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7358 ;
7359 046226 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7360 046232 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7361 046240 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
7362 046244 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7363 046252 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
7364 046260 016037 000012 001170 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7365 046266 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
7366 046274 042737 100100 001164 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7367 046302 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
7368 046310 016037 000012 001172 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7369 046316 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
7370 046324 042737 100100 001166 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7371 046332 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7372 046340 001006 BNE 68$;BR IF NOT
7373 046342 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7374 046346 001045 BNE 70$;BR IF NOT
7375 046350 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7376 046352 000137 046536 JMP 72$;BYPASS THE REST OF THE CHECKS
7377 046356 013737 001170 001126 68$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7378 046364 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7379 046372 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
7380 046400 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7381 046404 001414 BEQ 69$;BR IF ZERO
7382 046406 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7383 046414 013737 001172 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7384 046422 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
7385 046430 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7386 046434 001012 BNE 70$;BR IF NOT
7387 046436 012737 177777 001250 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7388 046444 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
7389 046452 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
7390 046460 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
7391 046462 013737 001170 001126 70$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7392 046470 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7393 046476 023737 001124 001170 CMP $GDDAT,STMP2 ;ALL BITS OK ?
7394 046504 001401 BEQ 71$;BR IF OK FROM PORT A.
7395 046506 104007 ERROR 7 ;REPORT ERROR
7396 046510 013737 001172 001126 71$: MOV STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7397 046516 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
7398 046524 023737 001124 001172 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
7399 046532 001401 BEQ 72$;BR IF OK
7400 046534 104007 ERROR 7 ;REPORT ERROR
7401 046536 000240 72$: NOP
7402 046540 000004 SCOPE ;LOOP ?
7403 ;
7404 ;:*****

```

```

7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420 046542
7421 046542 005737 001274
7422 046546 001406
7423 046550 100002
7424 046552 000137 002602
7425 046556 012737 177777 001274
7426 046564 112737 000040 001102
7427 046572 012737 046614 001106
7428 046600 012737 046614 001110
7429 046606 012737 000004 001176
7430 046614 012706 001100
7431
7432
7433
7434 046620 113760 001224 000010
7435 046626 005060 000012
7436 046632 012760 000011 000000
7437 046640 012760 000013 000000
7438 046646 113760 001226 000010
7439 046654 005060 000012
7440 046660 012760 000011 000000
7441 046666 012760 000013 000000
7442 046674 113760 001226 000010
7443 046702 013737 001226 001234
7444 046710 013737 001226 001236
7445
7446
7447
7448
7449 046716 012760 000007 000000
7450
7451
7452
7453
7454 046724 032760 000200 000012
7455 046732 001774
7456
7457
7458
7459
7460 046734 005037 001244

```

```

; *TEST 40 TEST PORT 'B' ATTENTION AFTER A COMMAND
; *
; *TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
; * COMMAND.
; *
; * A. ISSUE A RECALIBRATE COMMAND THROUGH PORT '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.
; *
; *****
; ST40:
; 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 #40,$STSTNM ;TEST NUMBER
; MOV #TEST40,$LPADR ;LOAD LOOP ON TEST ADDRESS
; MOV #TEST40,$LPERR ;LOAD LOOP ON ERROR ADDRESS
; MOV #4,$TIMES ;DO 4 ITERATIONS
; TEST40: MOV #STACK,$SP ;LOAD THE STACK POINTER
;
; ;CLEAR ATTENTION BITS FOR BOTH PORTS
; MOVB PORTA,$RPCS2($RO) ;SELECT PORT #A
; CLR RPDS1($RO) ;SEIZE THE DRIVE
; MOV #11,$RPCS1($RO) ;ISSUE DRIVE CLEAR
; MOV #13,$RPCS1($RO) ;RELEASE THE DRIVE
; MOVB PORTB,$RPCS2($RO) ;SELECT PORT #B
; CLR RPDS1($RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
; MOV #11,$RPCS1($RO) ;ISSUE DRIVE CLEAR
; MOV #13,$RPCS1($RO) ;RELEASE THE DRIVE
; MOVB PORTB,$RPCS2($RO) ;SELECT PORT B
; MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
; MOV PORTB,$SEIZPT ;'SEIZED' PORT ADDRESS
; *****
; ;DO A RECALIBRATE THROUGH PORT B
; MOV #7,$RPCS1($RO) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B
; *****
; ;WAIT FOR DRIVE TO FINISH
; BIT #DRY,$RPDS1($RO) ;WAIT FOR DRIVE TO FINISH
; BEQ .-6 ;BR IF NOT FINISHED
; *****
; ;CONFIRM THAT ATTENTION IS SET FOR PORT B
; CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR

```

```

7461 046740 016037 000012 001126
7462 046746 012737 000012 001122
7463 046754 060037 001122
7464 046760 012737 100000 001124
7465 046766 013737 001126 001164
7466 046774 042737 077777 001164
7467 047002 023737 001124 001164
7468 047010 001414
7469 047012 013737 001126 001174
7470 047020 042737 100000 001174
7471 047026 053737 001174 001124
7472 047034 104032
7473 047036 005137 001244
7474 047042 000240
7475 047044 113760 001224 000010
7476 047052 013737 001224 001234
7477
7478
7479
7480
7481 047060 005037 001244
7482 047064 016037 000012 001126
7483 047072 012737 000012 001122
7484 047100 060037 001122
7485 047104 005037 001124
7486 047110 013737 001126 001164
7487 047116 042737 077777 001164
7488 047124 023737 001124 001164
7489 047132 001414
7490 047134 013737 001126 001174
7491 047142 042737 100000 001174
7492 047150 053737 001174 001124
7493 047156 104032
7494 047160 005137 001244
7495 047164 000240
7496
7497
7498
7499
7500
7501 047166 113760 001226 000010
7502 047174 013737 001226 001234
7503 047202 012760 000013 000000
7504
7505
7506
7507 047210 005037 001250
7508 047214 012737 000012 001122
7509 047222 060037 001122
7510 047226 012737 011700 001124
7511 047234 113760 001224 000010
7512 047242 016037 000012 001170
7513 047250 013737 001170 001164
7514 047256 042737 100100 001164
7515 047264 113760 001226 000010
7516 047272 016037 000012 001172

```

```

MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #+CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 64$;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 32 ;TYPE MESSAGE 32
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
64$: MOVB PORTA, RPCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

;*****
;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

```

```

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #+CATA, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 66$;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 32 ;TYPE MESSAGE 32
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
66$:

```

```

;*****
;RELEASE THE DRIVE FROM PORT B

```

```

MOVB PORTB, RPCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13, RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1, $B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO, $B0ADR ;ADD THE I/O BASE ADDRESS
MOV #MOL:PGM:DPR:DRY:VV, $GDDAT ;COMPARISON CONSTANT
MOVB PORTA, RPCS2(RO) ;SELECT PORT A.
MOV RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
BIC #ATA:VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB, RPCS2(RO) ;SELECT PORT B.
MOV RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

|      |        |        |        |        |       |       |                 |                                                    |
|------|--------|--------|--------|--------|-------|-------|-----------------|----------------------------------------------------|
| 7517 | 047300 | 013737 | 001172 | 001166 |       | MOV   | \$TMP3,\$TMP1   | ;COPY IT INTO '\$TMP1'                             |
| 7518 | 047306 | 042737 | 100100 | 001166 |       | BIC   | #ATA:VV,\$TMP1  | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 7519 | 047314 | 023737 | 001164 | 001166 |       | CMP   | \$TMP0,\$TMP1   | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 7520 | 047322 | 001006 |        |        |       | BNE   | 68\$            | ;BR IF NOT                                         |
| 7521 | 047324 | 005737 | 001164 |        |       | TST   | \$TMP0          | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 7522 | 047330 | 001045 |        |        |       | BNE   | 70\$            | ;BR IF NOT                                         |
| 7523 | 047332 | 104046 |        |        |       | ERROR | 46              | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 7524 | 047334 | 000137 | 047520 |        |       | JMP   | 72\$            | ;BYPASS THE REST OF THE CHECKS                     |
| 7525 | 047340 | 013737 | 001170 | 001126 | 68\$: | MOV   | \$TMP2,\$BDDAT  | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 7526 | 047346 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR     | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 7527 | 047354 | 113760 | 001226 | 000010 |       | MOVB  | PORTB,RPCS2(RO) | ;SELECT PORT B.                                    |
| 7528 | 047362 | 005737 | 001164 |        |       | TST   | \$TMP0          | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 7529 | 047366 | 001414 |        |        |       | BEQ   | 69\$            | ;BR IF ZERO                                        |
| 7530 | 047370 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR     | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 7531 | 047376 | 013737 | 001172 | 001126 |       | MOV   | \$TMP3,\$BDDAT  | ; 'BAD DATA' FOR ERROR TYPE OUT                    |
| 7532 | 047404 | 113760 | 001224 | 000010 |       | MOVB  | PORTA,RPCS2(RO) | ;SELECT PORT A.                                    |
| 7533 | 047412 | 005737 | 001166 |        |       | TST   | \$TMP1          | ;SEE IF STATUS EQ ZERO FROM PORT B.                |
| 7534 | 047416 | 001012 |        |        |       | BNE   | 70\$            | ;BR IF NOT                                         |
| 7535 | 047420 | 012737 | 177777 | 001250 | 69\$: | MOV   | #-1,RELERR      | ;SET 'RELEASE ERROR' INDICATOR                     |
| 7536 | 047426 | 012760 | 000011 | 000000 |       | MOV   | #11,RPCS1(RO)   | ;CLEAR THE DRIVE                                   |
| 7537 | 047434 | 012760 | 000013 | 000000 |       | MOV   | #13,RPCS1(RO)   | ;RELEASE THE DRIVE                                 |
| 7538 | 047442 | 104026 |        |        |       | ERROR | 26              | ;TYPE ERROR MESSAGE 26                             |
| 7539 | 047444 | 013737 | 001170 | 001126 | 70\$: | MOV   | \$TMP2,\$BDDAT  | ;LOOK FOR BIT FAILURES WHEN RPDS1 READ             |
| 7540 | 047452 | 013737 | 001224 | 001234 |       | MOV   | PORTA,PTNBR     | ;CHANGE PORT NUMBER                                |
| 7541 | 047460 | 023737 | 001124 | 001170 |       | CMP   | \$GDDAT,\$TMP2  | ;ALL BITS OK ?                                     |
| 7542 | 047466 | 001401 |        |        |       | BEQ   | 71\$            | ;BR IF OK FROM PORT A.                             |
| 7543 | 047470 | 104007 |        |        |       | ERROR | 7               | ;REPORT ERROR                                      |
| 7544 | 047472 | 013737 | 001172 | 001126 | 71\$: | MOV   | \$TMP3,\$BDDAT  | ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.       |
| 7545 | 047500 | 013737 | 001226 | 001234 |       | MOV   | PORTB,PTNBR     | ;CHANGE PORT NUMBER                                |
| 7546 | 047506 | 023737 | 001124 | 001172 |       | CMP   | \$GDDAT,\$TMP3  | ;SEE IF READ OK FROM PORT B.                       |
| 7547 | 047514 | 001401 |        |        |       | BEQ   | 72\$            | ;BR IF OK                                          |
| 7548 | 047516 | 104007 |        |        |       | ERROR | 7               | ;REPORT ERROR                                      |
| 7549 | 047520 | 000240 |        |        | 72\$: | NOP   |                 |                                                    |
| 7550 | 047522 | 000004 |        |        |       | SCOPE |                 | ;LOOP ?                                            |

7551  
7552  
7553  
7554  
7555  
7556  
7557  
7558  
7559  
7560  
7561  
7562  
7563  
7564  
7565  
7566  
7567  
7568  
7569  
7570  
7571  
7572

```

*TEST 41 TEST PORT INTERACTION FROM PORT 'A'
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
*
* C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT
* 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
*
* E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT
* PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS
* SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
* SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.

```

```

7573
7574
7575
7576
7577
7578 047524
7579 047524 005737 001274
7580 047530 001406
7581 047532 100002
7582 047534 000137 002602
7583 047540 012737 177777 001274
7584 047546 112737 000041 001102
7585 047554 012737 047576 001106
7586 047562 012737 047576 001110
7587 047570 012737 007640 001176
7588 047576 012706 001100
7589
7590
7591
7592 047602 113760 001224 000010
7593 047610 005060 000012
7594 047614 012760 000011 000000
7595 047622 012760 000013 000000
7596 047630 113760 001226 000010
7597 047636 005060 000012
7598 047642 012760 000011 000000
7599 047650 012760 000013 000000
7600
7601
7602
7603 047656 113760 001224 000010
7604 047664 013737 001224 001236
7605 047672 005060 000012
7606 047676 013737 001226 001240
7607 047704 012760 177777 000014
7608 047712 012760 177777 000040
7609 047720 012760 177777 000042
7610 047726 113760 001226 000010
7611 047734 013737 001226 001234
7612 047742 004737 050600
7613 047746 113760 001224 000010
7614 047754 013737 001224 001234
7615 047762 005060 000042
7616 047766 005060 000040
7617 047772 005060 000014
7618 047776 013760 001232 000016
7619 050004 113760 001226 000010
7620 050012 013737 001226 001234
7621 050020 012760 177777 000014
7622 050026 012760 177777 000040
7623 050034 012760 177777 000042
7624 050042 113760 001224 000010
7625 050050 013737 001224 001234
7626 050056 004737 050600
7627
7628

```

```

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

TST41:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$;BR IF NOT
BPL 1$;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #41,$STNM ;TEST NUMBER
MOV #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST41: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT A
MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT A
MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT A
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
JSR PC,TST41B ;CHECK THE REGISTERS THROUGH PORT B
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR RPER3(RO) ;CLEAR RPER3 ON PORT A
CLR RPER2(RO) ;CLEAR RPER2 ON PORT A
CLR RPER1(RO) ;CLEAR RPER1 ON PORT A
MOV ASR1,RPAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT A
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT B
MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT B
MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT B
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
JSR PC,TST41B ;CHECK THE REGISTERS THROUGH PORT A

;RELEASE THE DRIVE FROM PORT A

```

```

7629
7630 050062 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
7631 050070 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7632 050076 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
7633
7634 ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
7635
7636 050104 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
7637 050110 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7638 050116 012737 000012 001122 MOV #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
7639 050124 060037 001122 ADD RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
7640 050130 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
7641 050136 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7642 050144 016037 000012 001164 MOV RPDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
7643 050152 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
7644 050160 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7645 050166 016037 000012 001126 MOV RPDS1(RO),$BDAT ;DRIVE STATUS FROM PORT A
7646 050174 001404 BEQ 66$;BR IF STATUS FROM PORT A ZERO
7647 050176 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO ?
7648 050202 001401 BEQ 66$;BR IF ZERO
7649 050204 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
7650 050206 013737 001164 001126 66$: MOV $TMP0,$BDAT ;CHECK STATUS FROM PORT B
7651 050214 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
7652 050222 023737 001124 001126 CMP $GDDAT,$BDAT ;COMPARE WITH CONSTANT
7653 050230 001401 BEQ 67$;BR IF OK
7654 050232 104027 ERROR 27 ;REPORT REGISTER ERROR
7655 050234 000240 NOP
7656
7657 ;RELEASE THE DRIVE FROM PORT B
7658
7659 050236 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
7660 050244 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7661 050252 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
7662
7663 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7664
7665 050260 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
7666 050264 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7667 050272 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
7668 050276 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
7669 050304 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
7670 050312 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7671 050320 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
7672 050326 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7673 050334 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
7674 050342 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7675 050350 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
7676 050356 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7677 050364 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7678 050372 001006 BNE 68$;BR IF NOT
7679 050374 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7680 050400 001045 BNE 70$;BR IF NOT
7681 050402 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7682 050404 000137 050570 JMP 72$;BYPASS THE REST OF THE CHECKS
7683 050410 013737 001170 001126 68$: MOV $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7684 050416 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

7685 050424 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
7686 050432 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
7687 050436 001414 69$ BEQ 69$;BR IF ZERO
7688 050440 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7689 050446 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
7690 050454 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
7691 050462 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
7692 050466 001012 70$ BNE 70$;BR IF NOT
7693 050470 012737 177777 001250 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
7694 050476 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
7695 050504 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
7696 050512 104026 70$: ERROR 26 ;TYPE ERROR MESSAGE 26
7697 050514 013737 001170 001126 MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7698 050522 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7699 050530 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
7700 050536 001401 71$ BEQ 71$;BR IF OK FROM PORT A.
7701 050540 104007 71$: ERROR 7 ;REPORT ERROR
7702 050542 013737 001172 001126 MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7703 050550 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
7704 050556 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
7705 050564 001401 72$ BEQ 72$;BR IF OK
7706 050566 104007 72$: ERROR 7 ;REPORT ERROR
7707 050570 000240 72$: NOP
7708 050572 000004 ;LOOP ?
7709 050574 000137 051132 JMP TST42 ;GO TO THE NEXT TEST
7710
7711 ;CHECK THE REGISTERS ON THE SELECTED PORT
7712
7713 050600 TST41B:
7714 050600 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7715 050604 016037 000014 001126 MOV RPER1(RO),$BDDAT ;GET CONTENTS OF RPER1
7716 050612 012737 000014 001122 MOV #RPER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7717 050620 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
7718 050624 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7719 050630 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7720 050636 001403 64$ BEQ 64$;BR IF OK
7721 050640 104006 64$: ERROR 6 ;TYPE MESSAGE 6
7722 050642 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7723 050646 016037 000000 001126 64$: MOV RPCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
7724 050654 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
7725 050662 060037 001122 ADD RO,$B0ADR ;ADDRESS BASE
7726 050666 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
7727 050674 001404 65$ BEQ 65$;BR IF NOT
7728 050676 104011 65$: ERROR 11 ;REPORT THE ERROR
7729 050700 012760 040000 000000 MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
7730 050706 000240 65$: NOP
7731 050710 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7732 050714 016037 000040 001126 MOV RPER2(RO),$BDDAT ;GET CONTENTS OF RPER2
7733 050722 012737 000040 001122 MOV #RPER2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7734 050730 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
7735 050734 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7736 050740 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7737 050746 001403 66$ BEQ 66$;BR IF OK
7738 050750 104006 66$: ERROR 6 ;TYPE MESSAGE 6
7739 050752 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7740 050756 016037 000000 001126 66$: MOV RPCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1

```

```

7741 050764 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
7742 050772 060037 001122 ADD R0,$B0ADR ;ADDRESS BASE
7743 050776 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
7744 051004 001404 BEQ 67$;BR IF NOT
7745 051006 104011 ERROR 11 ;REPORT THE ERROR
7746 051010 012760 040000 000000 MOV #TRE,RPCS1(R0) ;CLEAR 'MCPE'
7747 051016 000240 NOP
7748 051020 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7749 051024 016037 000042 001126 MOV RPER3(R0),$BDDAT ;GET CONTENTS OF RPER3
7750 051032 012737 000042 001122 MOV #RPER3,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7751 051040 060037 001122 ADD R0,$B0ADR ;ADD RH11 BASE ADDRESS
7752 051044 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7753 051050 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7754 051056 001403 BEQ 68$;BR IF OK
7755 051060 104006 ERROR 6 ;TYPE MESSAGE 6
7756 051062 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7757 051066 016037 000000 001126 MOV RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHCS1
7758 051074 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
7759 051102 060037 001122 ADD R0,$B0ADR ;ADDRESS BASE
7760 051106 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
7761 051114 001404 BEQ 69$;BR IF NOT
7762 051116 104011 ERROR 11 ;REPORT THE ERROR
7763 051120 012760 040000 000000 MOV #TRE,RPCS1(R0) ;CLEAR 'MCPE'
7764 051126 000240 NOP
7765 051130 000207 RTS PC ;RETURN

```

```

*TEST 42 TEST PORT INTERACTION FROM PORT 'B'
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
* B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
*
* C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT
* 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
*
* E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT
* PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
* F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS
* SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
* SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
*
* G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*

```

```

7791
7792 051132 TST42: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
7793 051132 005737 001274 BEQ 2$;BR IF NOT
7794 051136 001406 BPL 1$;BR IF JUST ENTERED TEST
7795 051140 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
7796 051142 000137 002602

```



|      |        |        |        |        |         |      |                                 |                                            |
|------|--------|--------|--------|--------|---------|------|---------------------------------|--------------------------------------------|
| 7797 | 051146 | 012737 | 177777 | 001274 | 1S:     | MOV  | #-1,KYBCTL                      | ;SET SINGLE TEST INDICATOR                 |
| 7798 | 051154 | 112737 | 000042 | 001102 | 2S:     | MOVB | #42,\$TSTNM                     | ;TEST NUMBER                               |
| 7799 | 051162 | 012737 | 051204 | 001106 |         | MOV  | #TEST42,\$LPADR                 | ;LOAD LOOP ON TEST ADDRESS                 |
| 7800 | 051170 | 012737 | 051204 | 001110 |         | MOV  | #TEST42,\$LPERR                 | ;LOAD LOOP ON ERROR ADDRESS                |
| 7801 | 051176 | 012737 | 007640 | 001176 |         | MOV  | #4000,\$TIMES                   | ;DO 4000. ITERATIONS                       |
| 7802 | 051204 | 012706 | 001100 |        | TEST42: | MOV  | #STACK,SP                       | ;LOAD THE STACK POINTER                    |
| 7803 |        |        |        |        |         |      |                                 |                                            |
| 7804 |        |        |        |        |         |      |                                 |                                            |
| 7805 |        |        |        |        |         |      |                                 |                                            |
| 7806 | 051210 | 113760 | 001224 | 000010 |         | MOVB | PORTA,RPCS2(RO)                 | ;SELECT PORT #A                            |
| 7807 | 051216 | 005060 | 000012 |        |         | CLR  | RPDS1(RO)                       | ;SEIZE THE DRIVE                           |
| 7808 | 051222 | 012760 | 000011 | 000000 |         | MOV  | #11,RPCS1(RO)                   | ;ISSUE DRIVE CLEAR                         |
| 7809 | 051230 | 012760 | 000013 | 000000 |         | MOV  | #13,RPCS1(RO)                   | ;RELEASE THE DRIVE                         |
| 7810 | 051236 | 113760 | 001226 | 000010 |         | MOVB | PORTB,RPCS2(RO)                 | ;SELECT PORT #B                            |
| 7811 | 051244 | 005060 | 000012 |        |         | CLR  | RPDS1(RO)                       | ;SEIZE THE DRIVE THROUGH PORT 'B'          |
| 7812 | 051250 | 012760 | 000011 | 000000 |         | MOV  | #11,RPCS1(RO)                   | ;ISSUE DRIVE CLEAR                         |
| 7813 | 051256 | 012760 | 000013 | 000000 |         | MOV  | #13,RPCS1(RO)                   | ;RELEASE THE DRIVE                         |
| 7814 |        |        |        |        |         |      |                                 |                                            |
| 7815 |        |        |        |        |         |      |                                 |                                            |
| 7816 |        |        |        |        |         |      |                                 |                                            |
| 7817 | 051264 | 113760 | 001226 | 000010 |         | MOVB | PORTB,RPCS2(RO)                 | ;SELECT PORT B                             |
| 7818 | 051272 | 013737 | 001226 | 001236 |         | MOV  | PORTB,SEIZPT                    | ;STORE SEIZING PORT'S ADDRESS              |
| 7819 | 051300 | 005060 | 000012 |        |         | CLR  | RPDS1(RO)                       | ;WRITE RPDS1                               |
| 7820 | 051304 | 013737 | 001224 | 001240 |         | MOV  | PORTA,OPPAT                     | ; 'OPPOSITE' PORT ADDRESS                  |
| 7821 | 051312 | 012760 | 177777 | 000014 |         | MOV  | #-1,RPER1(RO)                   | ;LOAD 1'S INTO RPER1 THROUGH PORT B        |
| 7822 | 051320 | 012760 | 177777 | 000040 |         | MOV  | #-1,RPER2(RO)                   | ;LOAD 1'S INTO RPER2 THROUGH PORT B        |
| 7823 | 051326 | 012760 | 177777 | 000042 |         | MOV  | #-1,RPER3(RO)                   | ;LOAD 1'S INTO RPER3 THROUGH PORT B        |
| 7824 | 051334 | 113760 | 001224 | 000010 |         | MOVB | PORTA,RPCS2(RO)                 | ;SELECT PORT A                             |
| 7825 | 051342 | 013737 | 001224 | 001234 |         | MOV  | PORTA,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 7826 | 051350 | 004737 | 052206 |        |         | JSR  | PC,TST42B                       | ;CHECK THE REGISTERS THROUGH PORT A        |
| 7827 | 051354 | 113760 | 001226 | 000010 |         | MOVB | PORTB,RPCS2(RO)                 | ;SELECT PORT B                             |
| 7828 | 051362 | 013737 | 001226 | 001234 |         | MOV  | PORTB,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 7829 | 051370 | 005060 | 000042 |        |         | CLR  | RPER3(RO)                       | ;CLEAR RPER3 ON PORT B                     |
| 7830 | 051374 | 005060 | 000040 |        |         | CLR  | RPER2(RO)                       | ;CLEAR RPER2 ON PORT B                     |
| 7831 | 051400 | 005060 | 000014 |        |         | CLR  | RPER1(RO)                       | ;CLEAR RPER1 ON PORT B                     |
| 7832 | 051404 | 013760 | 001232 | 000016 |         | MOV  | ASR1,RPAS(RO)                   | ;CLEAR THE ATTENTION BIT FOR PORT B        |
| 7833 | 051412 | 113760 | 001224 | 000010 |         | MOVB | PORTA,RPCS2(RO)                 | ;SELECT PORT A                             |
| 7834 | 051420 | 013737 | 001224 | 001234 |         | MOV  | PORTA,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 7835 | 051426 | 012760 | 177777 | 000014 |         | MOV  | #-1,RPER1(RO)                   | ;LOAD 1'S INTO RPER1 THROUGH PORT A        |
| 7836 | 051434 | 012760 | 177777 | 000040 |         | MOV  | #-1,RPER2(RO)                   | ;LOAD 1'S INTO RPER2 THROUGH PORT A        |
| 7837 | 051442 | 012760 | 177777 | 000042 |         | MOV  | #-1,RPER3(RO)                   | ;LOAD 1'S INTO RPER3 THROUGH PORT A        |
| 7838 | 051450 | 113760 | 001226 | 000010 |         | MOVB | PORTB,RPCS2(RO)                 | ;SELECT PORT B                             |
| 7839 | 051456 | 013737 | 001226 | 001234 |         | MOV  | PORTB,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 7840 | 051464 | 004737 | 052206 |        |         | JSR  | PC,TST42B                       | ;CHECK THE REGISTERS THROUGH PORT B        |
| 7841 |        |        |        |        |         |      |                                 |                                            |
| 7842 |        |        |        |        |         |      |                                 |                                            |
| 7843 |        |        |        |        |         |      |                                 |                                            |
| 7844 | 051470 | 113760 | 001226 | 000010 |         | MOVB | PORTB,RPCS2(RO)                 | ;SELECT PORT B                             |
| 7845 | 051476 | 013737 | 001226 | 001234 |         | MOV  | PORTB,PTNBR                     | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| 7846 | 051504 | 012760 | 000013 | 000000 |         | MOV  | #13,RPCS1(RO)                   | ;ISSUE RELEASE THROUGH PORT B              |
| 7847 |        |        |        |        |         |      |                                 |                                            |
| 7848 |        |        |        |        |         |      |                                 |                                            |
| 7849 |        |        |        |        |         |      |                                 |                                            |
| 7850 | 051512 | 005037 | 001250 |        |         | CLR  | RELERR                          | ;CLEAR 'RELEASE ERROR' INDICATOR           |
| 7851 | 051516 | 012737 | 111700 | 001124 |         | MOV  | #ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT | ;COMPARISON CONSTANT                       |
| 7852 | 051524 | 012737 | 000012 | 001122 |         | MOV  | #RPDS1,\$BDADR                  | ;REGISTER ADDRESS INCREMENT                |



```

7909 052112 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
7910 052120 104026 000000 000000 ERROR 26 ;TYPE ERROR MESSAGE 26
7911 052122 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7912 052130 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
7913 052136 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
7914 052144 001401 000000 000000 BEQ 71$;BR IF OK FROM PORT A.
7915 052146 104007 000000 000000 ERROR 7 ;REPORT ERROR
7916 052150 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7917 052156 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
7918 052164 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
7919 052172 001401 000000 000000 BEQ 72$;BR IF OK
7920 052174 104007 000000 000000 ERROR 7 ;REPORT ERROR
7921 052176 000240 000000 000000 72$: NOP
7922 052200 000004 000000 000000 SCOPE
7923 052202 000137 052540 000000 JMP TST43 ;LOOP ?
7924 ;GO TO THE NEXT TEST
7925 ;CHECK THE REGISTERS ON THE SELECTED PORT
7926
7927 052206 TST42B:
7928 052206 005037 001244 000000 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7929 052212 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
7930 052220 012737 000014 001122 MOV #RPER1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7931 052226 060037 001122 000000 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
7932 052232 005037 001124 000000 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7933 052236 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7934 052244 001403 000000 000000 BEQ 64$;BR IF OK
7935 052246 104006 000000 000000 ERROR 6 ;TYPE MESSAGE 6
7936 052250 005137 001244 000000 64$: COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7937 052254 016037 000000 001126 MOV RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
7938 052262 012737 000000 001122 MOV #RPCS1,$BDAOR ;FORM ADDRESS OF REGISTER
7939 052270 060037 001122 000000 ADD RO,$BDAOR ;ADDRESS BASE
7940 052274 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
7941 052302 001404 000000 000000 BEQ 65$;BR IF NOT
7942 052304 104011 000000 000000 ERROR 11 ;REPORT THE ERROR
7943 052306 012760 040000 000000 MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
7944 052314 000240 000000 000000 65$: NOP
7945 052316 005037 001244 000000 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7946 052322 016037 000040 001126 MOV RPER2(RO), $BDDAT ;GET CONTENTS OF RPER2
7947 052330 012737 000040 001122 MOV #RPER2,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7948 052336 060037 001122 000000 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
7949 052342 005037 001124 000000 CLR $GDDAT ;WHAT REGISTER SHOULD BE
7950 052346 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
7951 052354 001403 000000 000000 BEQ 66$;BR IF OK
7952 052356 104006 000000 000000 ERROR 6 ;TYPE MESSAGE 6
7953 052360 005137 001244 000000 66$: COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7954 052364 016037 000000 001126 MOV RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
7955 052372 012737 000000 001122 MOV #RPCS1,$BDAOR ;FORM ADDRESS OF REGISTER
7956 052400 060037 001122 000000 ADD RO,$BDAOR ;ADDRESS BASE
7957 052404 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
7958 052412 001404 000000 000000 BEQ 67$;BR IF NOT
7959 052414 104011 000000 000000 ERROR 11 ;REPORT THE ERROR
7960 052416 012760 040000 000000 MOV #TRE,RPCS1(RO) ;CLEAR 'MCPE'
7961 052424 000240 000000 000000 67$: NOP
7962 052426 005037 001244 000000 000000 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
7963 052432 016037 000042 001126 MOV RPER3(RO), $BDDAT ;GET CONTENTS OF RPER3
7964 052440 012737 000042 001122 MOV #RPER3,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

```

7965 052446 060037 001122 ADD R0,$B0ADR ;ADD RH11 BASE ADDRESS
7966 052452 005037 001124 CLR $G0DAT ;WHAT REGISTER SHOULD BE
7967 052456 023737 001124 001126 CMP $G0DAT,$B0DAT ;IS THE REGISTER OK ?
7968 052464 001403 BEQ 68$;BR IF OK
7969 052466 104006 ERROR 6 ;TYPE MESSAGE 6
7970 052470 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
7971 052474 016037 000000 001126 68$: MOV RPCS1(R0),$B0DAT ;GET THE CONTENTS OF RHCS1
7972 052502 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
7973 052510 060037 001122 ADD R0,$B0ADR ;ADDRESS BASE
7974 052514 032737 020000 001126 BIT #MCPE,$B0DAT ;IS 'MCPE' SET ?
7975 052522 001404 BEQ 69$;BR IF NOT
7976 052524 104011 ERROR 11 ;REPORT THE ERROR
7977 052526 012760 040000 000000 MOV #TRE,RPCS1(R0) ;CLEAR 'MCPE'
7978 052534 000240 69$: NOP
7979 052536 000207 RTS PC ;RETURN

```

```

*TEST 43 TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
*
*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
*
* A. SET THE ATTENTION BIT FOR PORT 'A'.
*
* B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.

```

```

7994
7995 052540 †ST43:
7996 052540 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
7997 052544 001406 BEQ 2$;BR IF NOT
7998 052546 100002 BPL 1$;BR IF JUST ENTERED TEST
7999 052550 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
8000 052554 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
8001 052562 112737 000043 001102 2$: MOVB #43,$STSTM ;TEST NUMBER
8002 052570 012737 052612 001106 MOV #TEST43,$LPADR ;LOAD LOOP ON TEST ADDRESS
8003 052576 012737 052612 001110 MOV #TEST43,$LPERR ;LOAD LOOP ON ERROR ADDRESS
8004 052604 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
8005 052612 012706 001100 TEST43: MOV #STACK,$P ;LOAD THE STACK POINTER
8006
8007 ;CLEAR ATTENTION BITS FOR BOTH PORTS
8008
8009 052616 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT #A
8010 052624 005060 000012 CLR RPDS1(R0) ;SEIZE THE DRIVE
8011 052630 012760 000011 000000 MOV #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
8012 052636 012760 000013 000000 MOV #13,RPCS1(R0) ;RELEASE THE DRIVE
8013 052644 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT #B
8014 052652 005060 000012 CLR RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
8015 052656 012760 000011 000000 MOV #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
8016 052664 012760 000013 000000 MOV #13,RPCS1(R0) ;RELEASE THE DRIVE
8017 052672 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
8018 052700 012760 177777 000014 MOV #-1,RPER1(R0) ;SET ERRORS TO FORCE ATTN BIT ON PORT A
8019 052706 005060 000014 CLR RPER1(R0) ;CLEAR THE ERRORS
8020 052712 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B

```

```

8021 052720 005760 000012 1$: TST RPDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
8022 052724 001775 BEQ 1$;BR IF STILL SEIZED BY PORT A
8023 052726 012737 000016 001122 MOV #RPAS,SBDADR ;FORM ADDRESS OF ATTN REG IF ERROR
8024 052734 060037 001122 ADD RO,SBDADR ;ADD THE ADDRESS BASE
8025 052740 013737 001232 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
8026 052746 013737 001232 001166 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
8027 052754 005137 001166 COM $TMP1 ;COMPLEMENT IT
8028 052760 012737 053014 001110 MOV #2$,SLPERR ;LOAD LOOP ON ERROR ADDRESS
8029 052766 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
8030 052774 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8031 053002 013737 001226 001236 MOV PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
8032 053010 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT B
8033 053014 016037 000016 001126 2$: MOV RPAS(RO),SBDAT ;GET THE CONTENTS OF THE ATTENTION REG
8034 053022 013737 001126 001164 MOV SBDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
8035 053030 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
8036 053036 023737 001124 001164 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
8037 053044 001401 BEQ 3$;BR IF SET
8038 053046 104053 ERROR 53 ;REPORT THE ERROR
8039 053050 3$:
8040
8041 ;RELEASE THE DRIVE FROM PORT B
8042
8043 053050 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
8044 053056 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8045 053064 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
8046
8047 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8048
8049 053072 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8050 053076 012737 000012 001122 MOV #RPDS1,SBDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8051 053104 060037 001122 ADD RO,SBDADR ;ADD THE I/O BASE ADDRESS
8052 053110 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8053 053116 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8054 053124 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8055 053132 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
8056 053140 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8057 053146 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8058 053154 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8059 053162 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
8060 053170 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8061 053176 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8062 053204 001006 BNE 64$;BR IF NOT
8063 053206 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8064 053212 001045 BNE 66$;BR IF NOT
8065 053214 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8066 053216 000137 053416 JMP 68$;BYPASS THE REST OF THE CHECKS
8067 053222 013737 001170 001126 64$: MOV $TMP2,$SBDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8068 053230 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8069 053236 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8070 053244 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8071 053250 001414 BEQ 65$;BR IF ZERO
8072 053252 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8073 053260 013737 001172 001126 MOV $TMP3,$SBDAT ;'BAD DATA' FOR ERROR TYPE OUT
8074 053266 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8075 053274 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8076 053300 001012 BNE 66$;BR IF NOT

```

```

8077 053302 012737 177777 001250 65$: MOV # -1,RELEA ;SET 'RELEASE ERROR' INDICATOR
8078 053310 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
8079 053316 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8080 053324 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
8081 053326 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8082 053334 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8083 053342 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
8084 053350 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
8085 053356 001401 BEQ 67$;BR IF OK FROM PORT A.
8086 053360 104007 ERROR 7 ;REPORT ERROR
8087 053362 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8088 053370 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8089 053376 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
8090 053404 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8091 053412 001401 BEQ 68$;BR IF OK
8092 053414 104007 ERROR 7 ;REPORT ERROR
8093 053416 000240 68$: NOP
8094 053420 000004 SCOPE ;LOOP ?

```

```

*TEST 44 TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
*
*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
*
* A. SET THE ATTENTION BIT FOR PORT 'B'.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.

```

```

8109 053422 TST44: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
8110 053422 005737 001274 BEQ 2$;BR IF NOT
8111 053426 001406 BPL 1$;BR IF JUST ENTERED TEST
8112 053430 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
8113 053432 000137 002602 1$: MOV # -1,KYBCTL ;SET SINGLE TEST INDICATOR
8114 053436 012737 177777 001274 2$: MOV #44,$STNM ;TEST NUMBER
8115 053444 112737 000044 001102 MOV #TEST44,$LPADR ;LOAD LOOP ON TEST ADDRESS
8116 053452 012737 053474 001106 MOV #TEST44,$LPERR ;LOAD LOOP ON ERROR ADDRESS
8117 053460 012737 053474 001110 MOV #25,$TIMES ;DO 25 ITERATIONS
8118 053466 012737 000031 001176 TEST44: MOV #STACK,SP ;LOAD THE STACK POINTER
8119 053474 012706 001100 ;CLEAR ATTENTION BITS FOR BOTH PORTS
8120
8121
8122
8123 053500 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
8124 053506 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
8125 053512 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
8126 053520 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8127 053526 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
8128 053534 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
8129 053540 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
8130 053546 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8131 053554 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
8132 053562 012760 177777 000014 MOV # -1,RPER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT B

```

|      |        |        |        |        |           |                             |                                                    |
|------|--------|--------|--------|--------|-----------|-----------------------------|----------------------------------------------------|
| 8133 | 053570 | 005060 | 000014 |        | CLR       | RPER1(RO)                   | ;CLEAR THE ERRORS                                  |
| 8134 | 053574 | 113760 | 001224 | 000010 | MOVB      | PORTA,RPCS2(RO)             | ;SELECT PORT A                                     |
| 8135 | 053602 | 005760 | 000012 |        | 1\$: TST  | RPDS1(RO)                   | ;WAIT FOR DRIVE TO RETURN TO NEUTRAL               |
| 8136 | 053606 | 001775 |        |        | BEQ       | 1\$                         | ;BR IF STILL SEIZED BY PORT B                      |
| 8137 | 053610 | 012737 | 000016 | 001122 | MOV       | #RPAS,\$BDADR               | ;FORM ADDRESS OF ATTN REG IF ERROR                 |
| 8138 | 053616 | 060037 | 001122 |        | ADD       | RO,\$BDADR                  | ;ADD THE ADDRESS BASE                              |
| 8139 | 053622 | 013737 | 001232 | 001124 | MOV       | ASR1,\$GDDAT                | ;GOOD DATA FOR ERROR MESSAGE                       |
| 8140 | 053630 | 013737 | 001232 | 001166 | MOV       | ASR1,\$TMP1                 | ;MAKE DATA COMPARE MASK                            |
| 8141 | 053636 | 005137 | 001166 |        | COM       | \$TMP1                      | ;COMPLEMENT IT                                     |
| 8142 | 053642 | 012737 | 053676 | 001110 | MOV       | #2\$,\$LPERR                | ;LOAD LOOP ON ERROR ADDRESS                        |
| 8143 | 053650 | 113760 | 001224 | 000010 | MOVB      | PORTA,RPCS2(RO)             | ;SELECT PORT A                                     |
| 8144 | 053656 | 013737 | 001224 | 001234 | MOV       | PORTA,PTNBR                 | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 8145 | 053664 | 013737 | 001224 | 001236 | MOV       | PORTA,SEIZPT                | ; 'SEIZED' PORT ADDRESS                            |
| 8146 | 053672 | 005060 | 000012 |        | CLR       | RPDS1(RO)                   | ;SEIZE THE DRIVE THROUGH PORT A                    |
| 8147 | 053676 | 016037 | 000016 | 001126 | 2\$: MOV  | RPAS(RO),\$BDDAT            | ;GET THE CONTENTS OF THE ATTENTION REG             |
| 8148 | 053704 | 013737 | 001126 | 001164 | MOV       | \$BDDAT,\$TMP0              | ;PUT CONTENTS INTO WORKING LOCATION                |
| 8149 | 053712 | 043737 | 001166 | 001164 | BIC       | \$TMP1,\$TMP0               | ;CLEAR OTHER BITS                                  |
| 8150 | 053720 | 023737 | 001124 | 001164 | CMP       | \$GDDAT,\$TMP0              | ;SEE IF ATTN BIT FOR DRIVE SET                     |
| 8151 | 053726 | 001401 |        |        | BEQ       | 3\$                         | ;BR IF SET                                         |
| 8152 | 053730 | 104053 |        |        | ERROR     | 53                          | ;REPORT THE ERROR                                  |
| 8153 | 053732 |        |        |        | 3\$:      |                             |                                                    |
| 8154 |        |        |        |        |           |                             |                                                    |
| 8155 |        |        |        |        |           |                             | ;RELEASE THE DRIVE FROM PORT A                     |
| 8156 |        |        |        |        |           |                             |                                                    |
| 8157 | 053732 | 113760 | 001224 | 000010 | MOVB      | PORTA,RPCS2(RO)             | ;SELECT PORT A                                     |
| 8158 | 053740 | 013737 | 001224 | 001234 | MOV       | PORTA,PTNBR                 | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 8159 | 053746 | 012760 | 000013 | 000000 | MOV       | #13,RPCS1(RO)               | ;ISSUE RELEASE THROUGH PORT A                      |
| 8160 |        |        |        |        |           |                             |                                                    |
| 8161 |        |        |        |        |           |                             | ;VERIFY THAT THE DRIVE IS IN NEUTRAL               |
| 8162 |        |        |        |        |           |                             |                                                    |
| 8163 | 053754 | 005037 | 001250 |        | CLR       | RELEERR                     | ;CLEAR THE 'RELEASE ERROR' INDICATOR               |
| 8164 | 053760 | 012737 | 000012 | 001122 | MOV       | #RPDS1,\$BDADR              | ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT             |
| 8165 | 053766 | 060037 | 001122 |        | ADD       | RO,\$BDADR                  | ;ADD THE I/O BASE ADDRESS                          |
| 8166 | 053772 | 012737 | 011700 | 001124 | MOV       | #MOL!PGM!DPR!DRY!VV,\$GDDAT | ;COMPARISON CONSTANT                               |
| 8167 | 054000 | 113760 | 001224 | 000010 | MOVB      | PORTA,RPCS2(RO)             | ;SELECT PORT A.                                    |
| 8168 | 054006 | 016037 | 000012 | 001170 | MOV       | RPDS1(RO),\$TMP2            | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 8169 | 054014 | 013737 | 001170 | 001164 | MOV       | \$TMP2,\$TMP0               | ;COPY IT INTO '\$TMP0'                             |
| 8170 | 054022 | 042737 | 100100 | 001164 | BIC       | #ATA!VV,\$TMP0              | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 8171 | 054030 | 113760 | 001226 | 000010 | MOVB      | PORTB,RPCS2(RO)             | ;SELECT PORT B.                                    |
| 8172 | 054036 | 016037 | 000012 | 001172 | MOV       | RPDS1(RO),\$TMP3            | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 8173 | 054044 | 013737 | 001172 | 001166 | MOV       | \$TMP3,\$TMP1               | ;COPY IT INTO '\$TMP1'                             |
| 8174 | 054052 | 042737 | 100100 | 001166 | BIC       | #ATA!VV,\$TMP1              | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 8175 | 054060 | 023737 | 001164 | 001166 | CMP       | \$TMP0,\$TMP1               | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 8176 | 054066 | 001006 |        |        | BNE       | 64\$                        | ;BR IF NOT                                         |
| 8177 | 054070 | 005737 | 001164 |        | TST       | \$TMP0                      | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 8178 | 054074 | 001045 |        |        | BNE       | 66\$                        | ;BR IF NOT                                         |
| 8179 | 054076 | 104046 |        |        | ERROR     | 46                          | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 8180 | 054100 | 000137 | 054300 |        | JMP       | 68\$                        | ;BYPASS THE REST OF THE CHECKS                     |
| 8181 | 054104 | 013737 | 001170 | 001126 | 64\$: MOV | \$TMP2,\$BDDAT              | ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 8182 | 054112 | 013737 | 001226 | 001234 | MOV       | PORTB,PTNBR                 | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 8183 | 054120 | 113760 | 001226 | 000010 | MOVB      | PORTB,RPCS2(RO)             | ;SELECT PORT B.                                    |
| 8184 | 054126 | 005737 | 001164 |        | TST       | \$TMP0                      | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 8185 | 054132 | 001414 |        |        | BEQ       | 65\$                        | ;BR IF ZERO                                        |
| 8186 | 054134 | 013737 | 001224 | 001234 | MOV       | PORTA,PTNBR                 | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 8187 | 054142 | 013737 | 001172 | 001126 | MOV       | \$TMP3,\$BDDAT              | ; 'BAD DATA' FOR ERROR TYPE OUT                    |
| 8188 | 054150 | 113760 | 001224 | 000010 | MOVB      | PORTA,RPCS2(RO)             | ;SELECT PORT A.                                    |

```

8189 054156 005737 001166 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8190 054162 001012 BNE 66$;BR IF NOT
8191 054164 012737 177777 001250 65$: MOV #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
8192 054172 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
8193 054200 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8194 054206 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
8195 054210 013737 001170 001126 66$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8196 054216 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8197 054224 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
8198 054232 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
8199 054240 001401 BEQ 67$;BR IF OK FROM PORT A.
8200 054242 104007 ERROR 7 ;REPORT ERROR
8201 054244 013737 001172 001126 67$: MOV STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8202 054252 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8203 054260 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
8204 054266 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8205 054274 001401 BEQ 68$;BR IF OK
8206 054276 104007 ERROR 7 ;REPORT ERROR
8207 054300 000240 68$: NOP ;LOOP ?
8208 054302 000004 SCOPE ;GO TO END OF TEST
8209 054304 000137 056224 JMP SEOP

```

\*\*\*\*\*

.SBTTL \*\*\* SPECIAL TESTS FOR THE M775 ('DP') BOARD \*\*\*

\*\*\*\*\*

\*\*\*\*\*

```

: *TEST 45 TEST NO TIMEOUT THROUGH PORT 'A'
: *
: *VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
: * SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
: *
: * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
: *
: * B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
: *
: * C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE
: * HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
: * SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
: *
: * D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
: * BEEN RELEASED.
: *
: * E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
: * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
: *
: *****

```

```

8239 054310 005737 001274 †ST45: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
8240 054310 001406 BEQ 2$;BR IF NOT
8241 054314 100002 BPL 1$;BR IF JUST ENTERED TEST
8242 054320 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
8243 054324 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR

```



```

8245 054332 112737 000045 001102 25: MOV #45,$STSTNM ;TEST NUMBER
8246 054340 012737 054362 001106 MOV #TEST45,$LPADR ;LOAD LOOP ON TEST ADDRESS
8247 054346 012737 054362 001110 MOV #TEST45,$LPERR ;LOAD LOOP ON ERROR ADDRESS
8248 054354 012737 000004 001176 MOV #4,$TIMES ;DO 4 ITERATIONS
8249 054362 012706 001100 TEST45: MOV #STACK,$SP ;LOAD THE STACK POINTER
8250 ;CLEAR ATTENTION BITS FOR BOTH PORTS
8251
8252
8253 054366 113760 001224 000010 MOV PORTA,$RPCS2($RO) ;SELECT PORT #A
8254 054374 005060 000012 000000 CLR $RPS1($RO) ;SEIZE THE DRIVE
8255 054400 012760 000011 000000 MOV #11,$RPCS1($RO) ;ISSUE DRIVE CLEAR
8256 054406 012760 000013 000000 MOV #13,$RPCS1($RO) ;RELEASE THE DRIVE
8257 054414 113760 001226 000010 MOV PORTB,$RPCS2($RO) ;SELECT PORT #B
8258 054422 005060 000012 000000 CLR $RPS1($RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
8259 054426 012760 000011 000000 MOV #11,$RPCS1($RO) ;ISSUE DRIVE CLEAR
8260 054434 012760 000013 000000 MOV #13,$RPCS1($RO) ;RELEASE THE DRIVE
8261
8262 ;*****
8263
8264 ;SEIZE THE DRIVE THROUGH PORT B
8265
8266 054442 113760 001226 000010 MOV PORTB,$RPCS2($RO) ;SELECT PORT B
8267 054450 013737 001226 001236 MOV PORTB,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
8268 054456 005060 000012 000000 CLR $RPS1($RO) ;WRITE $RPS1
8269 054462 013737 001224 001240 MOV PORTA,$OPPRT ;'OPPOSITE' PORT ADDRESS
8270 054470 113760 001224 000010 MOV PORTA,$RPCS2($RO) ;SELECT PORT A
8271 054476 013737 001224 001234 MOV PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8272
8273 ;*****
8274 ;SET REQUEST THROUGH PORT A
8275
8276 054504 005060 000012 000000 CLR $RPS1($RO) ;SET REQUEST FOR PORT A
8277 054510 113760 001226 000010 MOV PORTB,$RPCS2($RO) ;SELECT PORT B
8278 054516 013737 001226 001234 MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8279
8280 ;*****
8281 ;RELEASE THE DRIVE THROUGH PORT B
8282
8283 054524 012760 000013 000000 MOV #13,$RPCS1($RO) ;RELEASE DRIVE THROUGH PORT B
8284
8285 ;*****
8286 ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8287
8288 054532 013737 001260 001254 MOV TIMEAP,$WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8289
8290 ;*****
8291 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
8292
8293 054540 005037 001244 000000 CLR $CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8294 054544 016037 000012 001126 MOV $RPS1($RO),$BDDAT ;GET CONTENTS OF $RPS1
8295 054552 012737 000012 001122 MOV #RPS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8296 054560 060037 001122 000000 ADD $RO,$B0ADR ;ADD R#11 BASE ADDRESS
8297 054564 005037 001124 000000 CLR $GDDAT ;WHAT REGISTER SHOULD BE
8298 054570 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
8299 054576 001403 000000 000000 BEQ 66 ;BR IF OK
8300 054600 104031 000000 000000 ERROR 31 ;TYPE MESSAGE 31

```

```

8301 054602 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8302 054606 000240 66$: NOP
8303 054610 005737 001244 TST CKERR ;REGISTER OK ?
8304 054614 001402 BEQ .+6 ;BR IF OK
8305 054616 000137 055254 JMP 1$;BYPASS REST OF TEST IF NOT
8306 054622 005737 001254 TST WATCH ;WATCH EQUAL ZERO ?
8307 054626 001375 BNE .-4 ;BR IF NOT
8308
8309 ;:*****
8310 ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
8311
8312 054630 013737 001224 001234 MOV PORTA,PTNBR ;PORT NUMBER FOR TYPEOUT
8313 054636 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8314 054642 016037 000012 001126 MOV RPDS1(RO) $BDDAT ;GET CONTENTS OF RPDS1
8315 054650 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8316 054656 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
8317 054662 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
8318 054666 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
8319 054674 001403 BEQ 68$;BR IF OK
8320 054676 104035 ERROR 35 ;TYPE MESSAGE 35
8321 054700 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8322 054704 000240 68$: NOP
8323 054706 005737 001244 TST CKERR ;REGISTER OK ?
8324 054712 001402 BEQ .+6 ;BR IF OK
8325 054714 000137 055254 JMP 1$;BYPASS REST OF TEST IF NOT
8326
8327 ;:*****
8328 ;RELEASE THE DRIVE FROM PORT A
8329
8330
8331 054720 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A
8332 054726 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8333 054734 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
8334
8335 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8336
8337 054742 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8338 054746 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8339 054754 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
8340 054760 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8341 054766 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
8342 054774 016037 000012 001170 MOV RPDS1(RO) $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8343 055002 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
8344 055010 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8345 055016 113760 001226 000010 MOVVB PORTB,RPCS2(RO) ;SELECT PORT B.
8346 055024 016037 000012 001172 MOV RPDS1(RO) $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8347 055032 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
8348 055040 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8349 055046 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8350 055054 001006 BNE 70$;BR IF NOT
8351 055056 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8352 055062 001045 BNE 72$;BR IF NOT
8353 055064 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8354 055066 000137 055252 JMP 74$;BYPASS THE REST OF THE CHECKS
8355 055072 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8356 055100 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

8357 055106 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8358 055114 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8359 055120 001414 BEQ 71$;BR IF ZERO
8360 055122 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8361 055130 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8362 055136 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8363 055144 005737 001164 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8364 055150 001012 BNE 72$;BR IF NOT
8365 055152 012737 177777 001250 71$: MOV #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
8366 055160 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
8367 055166 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8368 055174 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
8369 055176 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8370 055204 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8371 055212 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
8372 055220 001401 BEQ 73$;BR IF OK FROM PORT A.
8373 055222 104007 ERROR 7 ;REPORT ERROR
8374 055224 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8375 055232 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8376 055240 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8377 055246 001401 BEQ 74$;BR IF OK
8378 055250 104007 ERROR 7 ;REPORT ERROR
8379 055252 000240 74$: NOP
8380
8381 055254 000004 1$: SCOPE ;LOOP ?
8382

```

```

8383 *****
8384 *TEST 46 TEST NO TIMEOUT THROUGH PORT 'B'
8385 *
8386 *VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
8387 *SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
8388 *
8389 * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
8390 *
8391 * B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
8392 *
8393 * C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE
8394 *HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
8395 *SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
8396 *
8397 * D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
8398 *BEEN RELEASED.
8399 *
8400 * E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
8401 *RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
8402 *
8403 *****

```

```

8404 055256
8405 055256 005737 001274 †ST46: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
8406 055262 001406 BEQ 2$;BR IF NOT
8407 055264 100002 BPL 1$;BR IF JUST ENTERED TEST
8408 055266 000137 002602 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
8409 055272 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
8410 055300 112737 000046 001102 2$: MOVB #46,$STNM ;TEST NUMBER
8411 055306 012737 055330 001106 MOV #TEST46,$LPADR ;LOAD LOOP ON TEST ADDRESS
8412 055314 012737 055330 001110 MOV #TEST46,$LPERR ;LOAD LOOP ON ERROR ADDRESS

```

```

0413 055322 012737 000004 001176 MOV #4,STIMES ;;DO 4 ITERATIONS
0414 055330 012706 001100 TEST46: MOV #STACK,SP ;;LOAD THE STACK POINTER
0415
0416 ;CLEAR ATTENTION BITS FOR BOTH PORTS
0417
0418 055334 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT #A
0419 055342 005060 000012 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
0420 055346 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
0421 055354 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
0422 055362 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT #B
0423 055370 005060 000012 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
0424 055374 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
0425 055402 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
0426
0427 ;*****
0428
0429 ;SEIZE THE DRIVE THROUGH PORT A
0430
0431 055410 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
0432 055416 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
0433 055424 005060 000012 000012 CLR RPDS1(RO) ;WRITE RPDS1
0434 055430 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
0435 055436 113760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT PORT B
0436 055444 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
0437
0438 ;*****
0439 ;SET REQUEST THROUGH PORT B
0440
0441 055452 005060 000012 000010 CLR RPDS1(RO) ;SET REQUEST FOR PORT B
0442 055456 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A
0443 055464 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
0444
0445 ;*****
0446 ;RELEASE THE DRIVE THROUGH PORT A
0447
0448 055472 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE DRIVE THROUGH PORT A
0449
0450 ;*****
0451 ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
0452
0453 055500 013737 001266 001254 MOV TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
0454
0455 ;*****
0456 ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
0457
0458 055506 005037 001244 001126 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
0459 055512 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
0460 055520 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
0461 055526 060037 001122 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
0462 055532 005037 001124 001126 CLR $GDDAT ;WHAT REGISTER SHOULD BE
0463 055536 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
0464 055544 001403 001124 001126 BEQ 66$;BR IF OK
0465 055546 104031 001244 001126 ERROR 31 ;TYPE MESSAGE 31
0466 055550 005137 001244 001126 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
0467 055554 000240 001244 001126 NOP
0468 055556 005737 001244 001126 TST CKERR ;REGISTER OK ?

```

```

8469 055562 001402 BEQ +6 ;BR IF OK
8470 055564 000137 056222 JMP 1$;BYPASS REST OF TEST IF NOT
8471 055570 005737 001254 TST WATCH ;WATCH EQUAL ZERO ?
8472 055574 001375 BNE -4 ;BR IF NOT
8473
8474 ;:*****
8475 ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
8476
8477 055576 013737 001226 001234 MOV PORTB,PTNBR ;PORT NUMBER FOR TYPEOUT
8478 055604 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
8479 055610 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
8480 055616 012737 000012 001122 MOV #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8481 055624 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
8482 055630 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
8483 055634 023737 001124 001126 CMP SGDDAT,SBDDAT ;IS THE REGISTER OK ?
8484 055642 001403 BEQ 68$;BR IF OK
8485 055644 104035 ERROR 35 ;TYPE MESSAGE 35
8486 055646 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
8487 055652 000240 68$: NOP ;REGISTER OK ?
8488 055654 005737 001244 TST CKERR ;BR IF OK
8489 055660 001402 BEQ +6 ;BYPASS REST OF TEST IF NOT
8490 055662 000137 056222 JMP 1$
8491
8492 ;:*****
8493 ;RELEASE THE DRIVE FROM PORT B
8494
8495
8496 055666 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
8497 055674 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8498 055702 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
8499
8500 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8501
8502 055710 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
8503 055714 012737 000012 001122 MOV #RPDS1,SBADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8504 055722 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
8505 055726 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
8506 055734 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
8507 055742 016037 000012 001170 MOV RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8508 055750 013737 001170 001164 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
8509 055756 042737 100100 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8510 055764 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8511 055772 016037 000012 001172 MOV RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8512 056000 013737 001172 001166 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
8513 056006 042737 100100 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8514 056014 023737 001164 001166 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8515 056022 001006 BNE 70$;BR IF NOT
8516 056024 005737 001164 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8517 056030 001045 BNE 72$;BR IF NOT
8518 056032 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8519 056034 000137 056220 JMP 74$;BYPASS THE REST OF THE CHECKS
8520 056040 013737 001170 001126 70$: MOV STMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8521 056046 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8522 056054 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
8523 056062 005737 001164 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
8524 056066 001414 BEQ 71$;BR IF ZERO

```

```

8525 056070 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8526 056076 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
8527 056104 113760 001224 000010 MOVVB PORTA,RPCS2(RO) ;SELECT PORT A.
8528 056112 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
8529 056116 001012 BNE 72$;BR IF NOT
8530 056120 012737 177777 001250 71$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
8531 056126 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
8532 056134 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
8533 056142 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
8534 056144 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8535 056152 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
8536 056160 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
8537 056166 001401 BEQ 73$;BR IF OK FROM PORT A.
8538 056170 104007 ERROR 7 ;REPORT ERROR
8539 056172 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8540 056200 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
8541 056206 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
8542 056214 001401 BEQ 74$;BR IF OK
8543 056216 104007 ERROR 7 ;REPORT ERROR
8544 056220 000240 NOP
8545
8546 056222 000004 1$: SCOPE ;LOOP ?
8547
8548 .SBTTL END OF PASS ROUTINE
8549
8550 ;*****
8551 ;*INCREMENT THE PASS NUMBER ($PASS)
8552 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
8553 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
8554 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
8555 ;*IF THERES A MONITOR GO TO IT
8556 ;*IF THERE ISN'T JUMP TO TSTIAA
8557
8558 056224 SEOP:
8559 056224 005737 001274 TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
8560 056230 001402 BEQ .+6 ;BR IF NOT
8561 056232 000137 002602 JMP EXEC ;RETURN TO KEYBOARD CONTROL
8562 056236 005037 001102 CLR $STNM ;ZERO THE TEST NUMBER
8563 056242 005037 001176 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
8564 056246 005237 001100 INC $PASS ;INCREMENT THE PASS NUMBER
8565 056252 042737 100000 001100 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
8566 056260 005327 DEC (PC)+ ;LOOP?
8567 056262 000001 SEOPCT: .WORD 1
8568 056264 003063 BGT $DOAGN ;YES
8569 056266 012737 MOV (PC)+,2(PC)+ ;RESTORE COUNTER
8570 056270 000001 SENDCT: .WORD 1
8571 056272 056262 TYPE 65$;TYPE ASCIZ STRING
8572 056274 104401 056302 BR 64$;GET OVER THE ASCIZ
8573 056300 000407 ;:65$: .ASCIZ <12><15>/END PASS #/
8574
8575 056320 64$:
8576 056320 013746 001100 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
8577
8578 056324 104405 TYPDS ;TYPE PASS NUMBER
8579 056326 104401 056334 GO TYPE--DECIMAL ASCII WITH SIGN
8580 056332 000421 TYPE 67$;TYPE ASCIZ STRING
 BR 66$;GET OVER THE ASCIZ

```

```

9581 ;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
9582 66$:
9583 056376 013746 001112 MOV SERTTL,-(SP) ;:SAVE SERTTL FOR TYPEOUT
9584 ;:TOTAL NUMBER OF ERRORS
9585 056402 104405 TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN
9586 056404 104401 001207 TYPE .SCLF ;:TYPE CARRIAGE RETURN, LINE FEED
9587 056410 005037 001112 CLR SERTTL ;:CLEAR ERROR TOTAL
9588 056414 013700 000042 SGET42: MOV @#42,R0 ;:GET MONITOR ADDRESS
9589 056420 001405 BEQ $DOAGN ;:BRANCH IF NO MONITOR
9590 056422 000005 RESET ;:CLEAR THE WORLD
9591 056424 004710 SENDAD: JSR PC,(R0) ;:GO TO MONITOR
9592 056426 000240 NOP ;:SAVE ROOM
9593 056430 000240 NOP ;:FOR
9594 056432 000240 NOP ;:ACT11
9595 056434 SDOAGN:
9596 056434 000137 JMP @PC+ ;:RETURN
9597 056436 003064 SRTNAD: .WORD TST1AA
9598 056440 377 000 SNULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
9599 056444
9600
9601 ;:*****
9602
9603 .SBTTL *** SUBROUTINES ***
9604
9605 ;:*****
9606
9607
9608 ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
9609 ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
9610
9611 056444 012737 056514 000004 CKCLK: MOV @CKCLK1,@ERRVEC ;:SET UP VECTOR FOR CLOCK CHECK
9612 056452 005037 000006 CLR @ERRVEC+2 ;:NEW PSW
9613 056456 005777 122530 TST @SLKCSR ;:CHECK FOR KW11-P
9614 056462 013701 001216 MOV SLPVEC,R1 ;:KW11-P VECTOR ADDRESS
9615 056466 012721 056576 MOV @CLOCK,(R1)+ ;:SET UP KW11-P VECTOR
9616 056472 012711 000300 MOV #300,(R1) ;:PSW - PRI 6
9617 056476 012777 177777 122510 MOV #-1,@SLKCSB ;:LOAD COUNTER BUFFER WITH 1'S
9618 056504 012777 000135 122500 MOV #135,@SLKCSR ;:SET CLOCK - CNT UP, 16MS, CONT INT
9619 056512 000425 BR CKCLK3
9620 056514 062706 000004 CKCLK1: ADD #4,SP ;:RESTORE THE STACK POINTER
9621 056520 012737 056556 000004 MOV @CKCLK2,@ERRVEC ;:CHANGE ERROR VECTOR TO CHECK FOR KW11-L
9622 056526 005777 122466 TST @SLKS ;:LOOK FOR KW11-L
9623 056532 013701 001222 MOV SLLVEC,R1 ;:KW11-L VECTOR ADDRESS
9624 056536 012721 056576 MOV @CLOCK,(R1)+ ;:SET UP KW11-L VECTOR
9625 056542 012711 000300 MOV #300,(R1) ;:PSW - PRI 6
9626 056546 012777 000100 122444 MOV #100,@SLKS ;:SET KW11-L INTERRUPT
9627 056554 000404 BR CKCLK3
9628 056556 062706 000004 CKCLK2: ADD #4,SP ;:RESTORE THE STACK POINTER
9629 056562 062716 000002 ADD #2,(SP) ;:INCREMENT RETURN, NO CLOCK
9630 056566 012737 000006 000004 CKCLK3: MOV @ERRVEC ;:RESTORE THE ERROR VECTOR
9631 056574 000207 RTS PC
9632
9633 ;ROUTINE TO COUNT CLOCK TICKS
9634
9635 056576 062737 000021 001252 CLOCK: ADD #17,TIME ;:ADD 17 MS TO ELAPSED TIME COUNTER
9636 056604 005737 001254 TST WATCH ;:IS WATCH ALREADY ZERO ?

```

```

8637 056610 001406 BEQ 1$;BR IF IT IS
8638 056612 162737 000021 001254 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
8639 056620 100002 BPL 1$;BR IF NOT MINUS
8640 056622 005037 001254 CLR WATCH ;CLEAR WATCH DOG COUNTER
8641 056626 000002 RTI ;RETURN
8642
8643 ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
8644
8645 056630 162706 000004 TOLER: SUB #4,SP ;SETUP STACK
8646 056634 016616 000004 MOV 4(SP), (SP) ;SAVE STACK
8647 056640 013546 MOV 2(R5), -(SP) ;GET TIME VALUE
8648 056642 011666 000004 MOV (SP), 4(SP) ;MOVE TIME VALUE
8649 056646 011666 000006 MOV (SP), 6(SP) ;MOVE VALUE AGAIN
8650 056652 006216 ASR (SP) ;DIVIDE BY 2
8651 056654 006216 ASR (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
8652 056656 061666 000004 ADD (SP), 4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
8653 056662 162666 000004 SUB (SP), 4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
8654 056666 000205 RTS R5 ;RETURN WITH TOLERANCES ON THE STACK
8655
8656 ;*****
8657
8658 .SBTTL 'SYSMAC' UTILITY ROUTINES
8659
8660 ;*****
8661
8662 .SBTTL SCOPE HANDLER ROUTINE
8663
8664 ;*****
8665 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8666 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
8667 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8668 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8669 ;*SW14=1 LOOP ON TEST
8670 ;*SW11=1 INHIBIT ITERATIONS
8671 ;*SW09=1 LOOP ON ERROR
8672 ;*CALL
8673 ;* SCOPE ;;SCOPE=IOT
8674
8675 $SCOPE:
8676 056670 104407 CKSWR
8677 056672 032777 040000 122240 1$: BIT #BIT14, $SWR ;:TEST FOR CHANGE IN SOFT-SWR
8678 056700 001101 BNE $OVER ;:LOOP ON PRESENT TEST?
8679 ;*****START OF CODE FOR THE XOR TESTER***** ;:YES IF SW14=1
8680 056702 000416 BR 6$;:IF RUNNING ON THE "XOR" TESTER CHANGE
8681 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
8682 056704 013746 000004 MOV 2#ERRVEC, -(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
8683 056710 012737 056730 000004 MOV #5$, 2#ERRVEC ;:SET FOR TIMEOUT
8684 056716 005737 177060 TST 2#177060 ;:TIME OUT ON XOR?
8685 056722 012637 000004 MOV (SP)+, 2#ERRVEC ;:RESTORE THE ERROR VECTOR
8686 056726 000453 BR $$VLAD ;:GO TO THE NEXT TEST
8687 056730 022626 5$: CMP (SP)+, (SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
8688 056732 012637 000004 MOV (SP)+, 2#ERRVEC ;:RESTORE THE ERROR VECTOR
8689 056736 000413 BR 7$;:LOOP ON THE PRESENT TEST
8690 056740 6$: ;*****END OF CODE FOR THE XOR TESTER*****
8691 056740 105737 001103 2$: TSTB $ERFLG ;:HAS AN ERROR OCCURRED?
8692 056744 001421 BEQ 3$;:BR IF NO

```



```

8693 056746 123737 001115 001103 CMPB SERMAX, SERFLG ;; MAX. ERRORS FOR THIS TEST OCCURRED?
8694 056754 101015 BHI 3$;; BR IF NO
8695 056756 032777 001000 122154 BIT #BIT09, @SWR ;; LOOP ON ERROR?
8696 056764 001404 BEQ 4$;; BR IF NO
8697 056766 013737 001110 001106 7$: MOV SLPERR, SLPADR ;; SET LOOP ADDRESS TO LAST SCOPE
8698 056774 000443 BR SOVER
8699 056776 105037 001103 4$: CLRB SERFLG ;; ZERO THE ERROR FLAG
8700 057002 005037 001176 CLR $TIMES ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
8701 057006 000415 BR 1$;; ESCAPE TO THE NEXT TEST
8702 057010 032777 004000 122122 3$: BIT #BIT11, @SWR ;; INHIBIT ITERATIONS?
8703 057016 001011 BNE 1$;; BR IF YES
8704 057020 005737 001100 TST $SPASS ;; IF FIRST PASS OF PROGRAM
8705 057024 001406 BEQ 1$;; INHIBIT ITERATIONS
8706 057026 005237 001104 INC $ICNT ;; INCREMENT ITERATION COUNT
8707 057032 023737 001176 001104 CMP $TIMES, $ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
8708 057040 002021 BGE SOVER ;; BR IF MORE ITERATION REQUIRED
8709 057042 012737 000001 001104 1$: MOV #1, $ICNT ;; REINITIALIZE THE ITERATION COUNTER
8710 057050 013737 057120 001176 MOV $SMXCNT, $TIMES ;; SET NUMBER OF ITERATIONS TO DO
8711 057056 105237 001102 $SVLAD: INCB $TSTNM ;; COUNT TEST NUMBERS
8712 057062 011637 001106 MOV (SP), SLPADR ;; SAVE SCOPE LOOP ADDRESS
8713 057066 011637 001110 MOV (SP), SLPERR ;; SAVE ERROR LOOP ADDRESS
8714 057072 005037 001200 CLR $ESCAPE ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
8715 057076 112737 000001 001115 MOVB #1, SERMAX ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8716 057104 013777 001102 122030 $OVER: MOV $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER
8717 057112 013716 001106 MOV SLPADR, (SP) ;; FUDGE RETURN ADDRESS
8718 057116 000002 RTI
8719 057120 000004 SMXCNT: 4 ;; FIXES PS
8720 .SBTTL ERROR HANDLER ROUTINE ;; MAX. NUMBER OF ITERATIONS
8721
8722 ;; *****
8723 ;; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
8724 ;; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8725 ;; *AND GO TO SERRTYP ON ERROR
8726 ;; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8727 ;; *SW15=1 HALT ON ERROR
8728 ;; *SW13=1 INHIBIT ERROR TYPEOUTS
8729 ;; *SW10=1 BELL ON ERROR
8730 ;; *CALL
8731 ;; * ERROR N ;; ERROR=EMT AND N=ERROR ITEM NUMBER
8732
8733 $ERROR:
8734 057122 104407 CKSWR
8735 057124 113737 001102 001242 MOVB $TSTNM, TSTNUM ;; TEST FOR CHANGE IN SOFT-SWR
8736 057132 105237 001103 7$: INCB SERFLG ;; SET THE ERROR FLAG
8737 057136 001775 BEQ 7$;; DON'T LET THE FLAG GO TO ZERO
8738 057140 013777 001102 121774 MOV $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
8739 057146 032777 002000 121764 BIT #BIT10, @SWR ;; BELL ON ERROR?
8740 057154 001402 BEQ 1$;; NO - SKIP
8741 057156 104401 001202 TYPE $BELL ;; RING BELL
8742 057162 005237 001112 1$: INC $ERTTL ;; COUNT THE NUMBER OF ERRORS
8743 057166 011637 001116 MOV (SP), $ERRPC ;; GET ADDRESS OF ERROR INSTRUCTION
8744 057172 162737 000002 001116 SUB #2, $ERRPC
8745 057200 117737 121712 001114 MOVB @ERRPC, $ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
8746 057206 032777 020000 121724 BIT #BIT13, @SWR ;; SKIP TYPEOUT IF SET
8747 057214 001004 BNE 20$;; SKIP TYPEOUTS
8748 057216 004737 057254 JSR PC, SERRTYP ;; GO TO USER ERROR ROUTINE

```

8749 057222 104401 001207  
8750 057226  
8751 057226 005777 121706  
8752 057232 100002  
8753 057234 000000  
8754 057236 104407  
8755 057240  
8756 057240 022737 056424 000042  
8757 057246 001001  
8758 057250 000000  
8759 057252  
8760 057252 000002

20\$: TYPE ,SCLRF  
2\$: TST @SWR ;:HALT ON ERROR  
BPL 3\$ ;:SKIP IF CONTINUE  
HALT ;:HALT ON ERROR!  
CKSWR ;:TEST FOR CHANGE IN SOFT-SWR  
3\$: CMP #SENDAD,@#42 ;:ACT-11 AUTO-ACCEPT?  
BNE 6\$ ;:BRANCH IF NO  
HALT ;:YES  
6\$: RTI ;:RETURN  
.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

8761  
8762  
8763  
8764  
8765  
8766  
8767

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

8768 057254  
8769 057254 104401 001207  
8770 057260 010046  
8771 057262 005000  
8772 057264 153700 001114  
8773 057270 001004  
8774  
8775 057272 013746 001116  
8776  
8777 057276 104402  
8778 057300 000445  
8779 057302 005300  
8780 057304 006300  
8781 057306 006300  
8782 057310 006300  
8783 057312 062700 001304  
8784 057316 012037 057326  
8785 057322 001404  
8786 057324 104401  
8787 057326 000000  
8788 057330 104401 001207  
8789 057334 012037 057344  
8790 057340 001404  
8791 057342 104401  
8792 057344 000000  
8793 057346 104401 001207  
8794 057352 010146  
8795 057354 012001  
8796 057356 001415  
8797 057360 012000  
8798 057362 105720  
8799 057364 001003  
8800 057366 013146  
8801 057370 104402  
8802 057372 000402  
8803 057374  
8804 057374 013146

SERRTYP:  
TYPE ,SCLRF ;:"CARRIAGE RETURN" & "LINE FEED"  
MOV RO,-(SP) ;:SAVE RO  
CLR RO ;:PICKUP THE ITEM INDEX  
BISB @#\$ITEMB,RO  
BNE 1\$ ;:IF ITEM NUMBER IS ZERO, JUST  
TYPE THE PC OF THE ERROR  
MOV \$ERRPC,-(SP) ;:SAVE \$ERRPC FOR TYPEOUT  
ERROR ADDRESS  
TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)  
BR 10\$ ;:GET OUT  
1\$: DEC RO ;:ADJUST THE INDEX SO THAT IT WILL  
ASL RO ;:WORK FOR THE ERROR TABLE  
ASL RO  
ASL RO  
ADD #SERRTB,RO ;:FORM TABLE POINTER  
MOV (RO)+,2\$ ;:PICKUP "ERROR MESSAGE" POINTER  
BEQ 3\$ ;:SKIP TYPEOUT IF NO POINTER  
TYPE ;:TYPE THE "ERROR MESSAGE"  
"ERROR MESSAGE" POINTER GOES HERE  
2\$: WORD 0 ;:"CARRIAGE RETURN" & "LINE FEED"  
TYPE ,SCLRF ;:PICKUP "DATA HEADER" POINTER  
3\$: MOV (RO)+,4\$ ;:SKIP TYPEOUT IF 0  
BEQ 5\$ ;:TYPE THE "DATA HEADER"  
"DATA HEADER" POINTER GOES HERE  
4\$: WORD 0 ;:"CARRIAGE RETURN" & "LINE FEED"  
TYPE ,SCLRF ;:SAVE R1  
5\$: MOV R1,-(SP) ;:PICKUP "DATA TABLE" POINTER  
MOV (RO)+,R1 ;:BR IF NO DATA TO BE TYPED  
BEQ 9\$ ;:PICKUP "DATA FORMAT" POINTER  
MOV (RO)+,RO ;:"OCTAL" OR "DECIMAL"  
6\$: TSTB (RO)+ ;:BR IF DECIMAL  
7\$: BNE 7\$ ;:SAVE @ (R1)+ FOR TYPEOUT  
MOV @ (R1)+,-(SP) ;:GO TYPE--OCTAL ASCII(ALL DIGITS)  
TYPOC  
BR 8\$  
7\$: MOV @ (R1)+,-(SP) ;:SAVE @ (R1)+ FOR TYPEOUT

```

8805 057376 104405
8806 057400 005711
8807 057402 001403
8808 057404 104401 057424
8809 057410 000764
8810
8811 057412 012601
8812 057414 012600
8813 057416 104401 001207
8814 057422 000207
8815 057424 020040 000
8816 057430
8817
8818
8819
8820
8821
8822
8823
8824
8825
8826
8827
8828
8829
8830
8831
8832
8833
8834 057430 105737 001157
8835 057434 100002
8836 057436 000000
8837 057440 000407
8838 057442 010046
8839 057444 017600 000002
8840 057450 112046
8841 057452 001005
8842 057454 005726
8843 057456 012600
8844 057460 062716 000002
8845 057464 000002
8846 057466 122716 000011
8847 057472 001430
8848 057474 122716 000200
8849 057500 001006
8850 057502 005726
8851 057504 104401
8852 057506 001207
8853 057510 105037 057644
8854 057514 000755
8855 057516 004737 057600
8856 057522 123726 001156
8857 057526 001350
8858 057530 013746 001154
8859
8860 057534 105366 000001

 TYPDS
8$: TST (R1) ;; GO TYPE--DECIMAL ASCII WITH SIGN
 BEQ 9$;; IS THERE ANOTHER NUMBER?
 TYPE 11$;; BR IF NO
 BR 6$;; TYPE TWO(2) SPACES
 ;; 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 3$;; HALT HERE IF NO TERMINAL
 BR 3$;; LEAVE
1$: MOV R0, -(SP) ;; SAVE R0
 MOV @2(SP),R0 ;; GET ADDRESS OF ASCIZ STRING
2$: MOVB (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$: ADD #2, (SP) ;; ADJUST RETURN PC
 RTI
4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
 BEQ 8$;;
 CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
 BNE 5$;;
 TST (SP)+ ;; POP <CR><LF> EQUIV
 ;; TYPE A CR AND LF
8$: CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
 BR 2$;; GET NEXT CHARACTER
5$: JSR PC, $TYPEPC ;; GO TYPE THIS CHARACTER
6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
 BNE 2$;; IF NO GO GET NEXT CHAR.
 MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
 ;; AND THE NULL CHAR.
7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?

```

```

8861 057540 002770 BLT 6$;; BR IF NO--GO POP THE NULL OFF OF STACK
8862 057542 004737 057600 JSR PC,$TYPEC ;; GO TYPE A NULL
8863 057546 105337 057644 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
8864 057552 000770 BR 7$;; LOOP
8865
8866 ;HORIZONTAL TAB PROCESSOR
8867
8868 057554 112716 000040 8$: MOVB #' (SP) ;; REPLACE TAB WITH SPACE
8869 057560 004737 057600 9$: JSR PC,$TYPEC ;; TYPE A SPACE
8870 057564 132737 000007 057644 BITB #',$CHARCNT ;; BRANCH IF NOT AT
8871 057572 001372 BNE 9$;; TAB STOP
8872 057574 005726 TST (SP)+ ;; POP SPACE OFF STACK
8873 057576 000724 BR 2$;; GET NEXT CHARACTER
8874 057600 105777 121344 $TYPEC: TSTB 2$STPS ;; WAIT UNTIL PRINTER IS READY
8875 057604 100375 BPL $TYPEC
8876 057606 116677 000002 121336 MOVB 2(SP),2$TPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.
8877 057614 122766 000015 000002 CMPB #CR,2(SP) ;; IS CHARACTER A CARRIAGE RETURN?
8878 057622 001003 BNE 1$;; BRANCH IF NO
8879 057624 105037 057644 CLRB $CHARCNT ;; YES--CLEAR CHARACTER COUNT
8880 057630 000406 BR $TYPEX ;; EXIT
8881 057632 122766 000012 000002 1$: CMPB #LF,2(SP) ;; IS CHARACTER A LINE FEED?
8882 057640 001402 BEQ $TYPEX ;; BRANCH IF YES
8883 057642 105227 INCB (PC)+ ;; COUNT THE CHARACTER
8884 057644 000000 $CHARCNT: .WORD 0 ;; CHARACTER COUNT STORAGE
8885 057646 000207 $TYPEX: RTS PC
8886
8887 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
8888
8889 ;*****
8890 ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
8891 ;OCTAL (ASCII) NUMBER AND TYPE IT.
8892 ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
8893 ;CALL:
8894 ;* MOV NUM,-(SP) ;; NUMBER TO BE TYPED
8895 ;* TYPOS ;; CALL FOR TYPEOUT
8896 ;* .BYTE N ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
8897 ;* .BYTE M ;; M=1 OR 0
8898 ;* ;; 1=TYPE LEADING ZEROS
8899 ;* ;; 0=SUPPRESS LEADING ZEROS
8900
8901 ;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
8902 ;$TYPOS OR $TYPOC
8903 ;CALL:
8904 ;* MOV NUM,-(SP) ;; NUMBER TO BE TYPED
8905 ;* TYPON ;; CALL FOR TYPEOUT
8906
8907 ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
8908 ;CALL:
8909 ;* MOV NUM,-(SP) ;; NUMBER TO BE TYPED
8910 ;* TYPOC ;; CALL FOR TYPEOUT
8911
8912 057650 017646 000000 $TYPOS: MOV 2(SP),-(SP) ;; PICKUP THE MODE
8913 057654 116637 000001 060073 MOVB 1(SP),$OFILL ;; LOAD ZERO FILL SWITCH
8914 057662 112637 060075 MOVB (SP)+,$OMODE+1 ;; NUMBER OF DIGITS TO TYPE
8915 057666 062716 000002 ADD #2,(SP) ;; ADJUST RETURN ADDRESS
8916 057672 000406 BR $TYPON

```

```

8917 057674 112737 000001 060073 $TYPOC: MOV B #1,$OFILL ;; SET THE ZERO FILL SWITCH
8918 057702 112737 000006 060075 MOV B #6,$SOMODE+1 ;; SET FOR SIX(6) DIGITS
8919 057710 112737 000005 060072 $TYPON: MOV B #5,$SOCNT ;; SET THE ITERATION COUNT
8920 057716 010346 MOV R3,-(SP) ;; SAVE R3
8921 057720 010446 MOV R4,-(SP) ;; SAVE R4
8922 057722 010546 MOV R5,-(SP) ;; SAVE R5
8923 057724 113704 060075 MOV B $SOMODE+1,R4 ;; GET THE NUMBER OF DIGITS TO TYPE
8924 057730 005404 NEG R4
8925 057732 062704 000006 ADD #6,R4 ;; SUBTRACT IT FOR MAX. ALLOWED
8926 057736 110437 060074 MOV B R4,$SOMODE ;; SAVE IT FOR USE
8927 057742 113704 060073 MOV B $OFILL,R4 ;; GET THE ZERO FILL SWITCH
8928 057746 016605 000012 MOV 12(SP),R5 ;; PICKUP THE INPUT NUMBER
8929 057752 005003 CLR R3 ;; CLEAR THE OUTPUT WORD
8930 057754 006105 1$: ROL R5 ;; ROTATE MSB INTO "C"
8931 057756 000404 BR 3$;; GO DO MSB
8932 057760 006105 2$: ROL R5 ;; FORM THIS DIGIT
8933 057762 006105 ROL R5
8934 057764 006105 ROL R5
8935 057766 010503 MOV R5,R3
8936 057770 006103 3$: ROL R3 ;; GET LSB OF THIS DIGIT
8937 057772 105337 060074 DECB $SOMODE ;; TYPE THIS DIGIT?
8938 057776 100016 BPL 7$;; BR IF NO
8939 060000 042703 177770 BIC #177770,R3 ;; GET RID OF JUNK
8940 060004 001002 BNE 4$;; TEST FOR 0
8941 060006 005704 TST R4 ;; SUPPRESS THIS 0?
8942 060010 001403 BEQ 5$;; BR IF YES
8943 060012 005204 4$: INC R4 ;; DON'T SUPPRESS ANYMORE 0'S
8944 060014 052703 000060 BIS #'0,R3 ;; MAKE THIS DIGIT ASCII
8945 060020 052703 000040 5$: BIS #' ,R3 ;; MAKE ASCII IF NOT ALREADY
8946 060024 110337 060070 MOV B R3,$S ;; SAVE FOR TYPING
8947 060030 104401 060070 TYPE #8$;; GO TYPE THIS DIGIT
8948 060034 105337 060072 7$: DECB $SOCNT ;; COUNT BY 1
8949 060040 003347 BGT 2$;; BR IF MORE TO DO
8950 060042 002402 BLT 6$;; BR IF DONE
8951 060044 005204 INC R4 ;; INSURE LAST DIGIT ISN'T A BLANK
8952 060046 000744 BR 2$;; GO DO THE LAST DIGIT
8953 060050 012605 6$: MOV (SP)+,R5 ;; RESTORE R5
8954 060052 012604 MOV (SP)+,R4 ;; RESTORE R4
8955 060054 012603 MOV (SP)+,R3 ;; RESTORE R3
8956 060056 016666 000002 000004 MOV 2(SP),4(SP) ;; SET THE STACK FOR RETURNING
8957 060064 012616 MOV (SP)+,(SP)
8958 060066 000002 RTI ;; RETURN
8959 060070 000 8$: .BYTE 0 ;; STORAGE FOR ASCII DIGIT
8960 060071 000 .BYTE 0 ;; TERMINATOR FOR TYPE ROUTINE
8961 060072 000 $SOCNT: .BYTE 0 ;; OCTAL DIGIT COUNTER
8962 060073 000 $OFILL: .BYTE 0 ;; ZERO FILL SWITCH
8963 060074 000000 $SOMODE: .WORD 0 ;; NUMBER OF DIGITS TO TYPE
8964 .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:

```

```

8965
8966
8967
8968
8969
8970
8971
8972

```

```

8973 ;* MOV NUM,-(SP) ;;PUT THE BINARY NUMBER ON THE STACK
8974 ;* TYPDS ;;GO TO THE ROUTINE
8975
8976 $TYPDS:
8977 060076 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
8978 060100 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
8979 060102 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
8980 060104 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
8981 060106 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
8982 060110 012746 020200 MOV #20200,-(SP) ;;SET BLANK SWITCH AND SIGN
8983 060114 016605 000020 MOV 20(SP),R5 ;;GET THE INPUT NUMBER
8984 060120 100004 BPL 1$;;BR IF INPUT IS POS.
8985 060122 005405 NEG R5 ;;MAKE THE BINARY NUMBER POS.
8986 060124 112766 000055 000001 MOVVB #'-,1(SP) ;;MAKE THE ASCII NUMBER NEG.
8987 060132 005000 CLR R0 ;;ZERO THE CONSTANTS INDEX
8988 060134 012703 060312 MOV #SDBLK,R3 ;;SETUP THE OUTPUT POINTER
8989 060140 112723 000040 MOVVB #' ,(R3)+ ;;SET THE FIRST CHARACTER TO A BLANK
8990 060144 005002 CLR R2 ;;CLEAR THE BCD NUMBER
8991 060146 016001 060302 MOV $DTBL(R0),R1 ;;GET THE CONSTANT
8992 060152 160105 3$: SUB R1,R5 ;;FORM THIS BCD DIGIT
8993 060154 002402 BLT 4$;;BR IF DONE
8994 060156 005202 INC R2 ;;INCREASE THE BCD DIGIT BY 1
8995 060160 000774 BR 3$
8996 060162 060105 4$: ADD R1,R5 ;;ADD BACK THE CONSTANT
8997 060164 005702 TST R2 ;;CHECK IF BCD DIGIT=0
8998 060166 001002 BNE 5$;;FALL THROUGH IF 0
8999 060170 105716 TSTB (SP) ;;STILL DOING LEADING 0'S?
9000 060172 100407 BMI 7$;;BR IF YES
9001 060174 106316 5$: ASLB (SP) ;;MSD?
9002 060176 103003 BCC 6$;;BR IF NO
9003 060200 116663 000001 177777 MOVVB 1(SP),-1(R3) ;;YES--SET THE SIGN
9004 060206 052702 000060 BIS #'0,R2 ;;MAKE THE BCD DIGIT ASCII
9005 060212 052702 000040 6$: BIS #' ,R2 ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9006 060216 110223 MOVVB R2,(R3)+ ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9007 060220 005720 TST (R0)+ ;;JUST INCREMENTING
9008 060222 020027 000010 CMP R0,#10 ;;CHECK THE TABLE INDEX
9009 060226 002746 BLT 2$;;GO DO THE NEXT DIGIT
9010 060230 003002 BGT 8$;;GO TO EXIT
9011 060232 010502 MOV R5,R2 ;;GET THE LSD
9012 060234 000764 BR 6$
9013 060236 105726 8$: TSTB (SP)+ ;;GO CHANGE TO ASCII
9014 060240 100003 BPL 9$;;WAS THE LSD THE FIRST NON-ZERO?
9015 060242 116663 177777 177776 MOVVB -1(SP),-2(R3) ;;BR IF NO
9016 060250 105013 9$: CLRB (R3) ;;YES--SET THE SIGN FOR TYPING
9017 060252 012605 MOV (SP)+,R5 ;;SET THE TERMINATOR
9018 060254 012603 MOV (SP)+,R3 ;;POP STACK INTO R5
9019 060256 012602 MOV (SP)+,R2 ;;POP STACK INTO R3
9020 060260 012601 MOV (SP)+,R1 ;;POP STACK INTO R2
9021 060262 012600 MOV (SP)+,R0 ;;POP STACK INTO R1
9022 060264 104401 060312 TYPE $SDBLK ;;POP STACK INTO R0
9023 060270 016666 000002 000004 MOV 2(SP),4(SP) ;;NOW TYPE THE NUMBER
9024 060276 012616 MOV (SP)+,(SP) ;;ADJUST THE STACK
9025 060300 000002 RTI
9026 060302 023420 $DTBL: 10000.
9027 060304 001750 1000.
9028 060306 000144 100.

```

```

9029 060310 000012
9030 060312 000004
9031
9032
9033
9034
9035 060322 000000
9036 060324 000000
9037 060326 000000
9038 060330 000001
9039 060331
9040 060332
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050 060332 005037 060322
9051 060336 012737 060330 060324
9052 060344 013737 060324 060326
9053 060352 012737 060402 000060
9054 060360 012737 000200 000062
9055 060366 005777 120554
9056 060372 012777 000100 120544
9057 060400 000207
9058
9059
9060
9061
9062
9063
9064 060402 117746 120540
9065 060406 042716 177600
9066 060412 021627 000007
9067 060416 001004
9068 060420 022737 000176 001140
9069 060426 001500
9070
9071 060430
9072 060430 022737 000001 060322
9073 060436 001004
9074 060440 104401 001202
9075 060444 005726
9076 060446 000451
9077 060450 021627 000023
9078 060454 001021
9079 060456 005077 120462
9080 060462 005726
9081 060464 105777 120454
9082 060470 100375
9083 060472 117746 120450
9084 060476 042716 177600

```

```

10.
$DBLK: .BLKW 4
.SBTTL TTY INPUT ROUTINE

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

: *TK INITIALIZE ROUTINE
: *THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
: *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 $TKQSRT,$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 $TKB, -(SP) ;:PICKUP THE CHARACTER
BIC #C177, (SP) ;:STRIP THE JUNK
1$: CMP (SP), #7 ;:IS IT A CONTROL G?
BNE 2$;:BRANCH IF NO
CMP #SWREG, SWR ;:IS SOFT-SWR SELECTED?
BEQ 6$;:GO TO SWR CHANGE

2$: CMP #1, $TKCNT ;:IS THE QUEUE FULL?
BNE 3$;:BRANCH IF NO
TYPE $BELL ;:RING THE TTY BELL
TST (SP)+ ;:CLEAN CHARACTER OFF OF STACK
BR 5$;:EXIT
3$: CMP (SP), #23 ;:IS IT A CONTROL-S?
BNE 32$;:BRANCH IF NO
CLR $TKS ;:DISABLE TTY KEYBOARD INTERRUPTS
TST (SP)+ ;:CLEAN CHAR OFF STACK
31$: TSTB $TKS ;:WAIT FOR A CHAR
BPL 31$;:LOOP UNTIL ITS THERE
MOVB $TKB, -(SP) ;:GET THE CHARACTER
BIC #C177, (SP) ;:MAKE IT 7-BIT ASCII

```

```

9085 060502 022627 000021 CMP (SP)+,#21 ;; IS IT A CONTROL-Q?
9086 060506 001366 BNE 31$;; BRANCH IF NO
9087 060510 012777 000100 120426 MOV #100,$STKS ;; REENABLE TTY KEYBOARD INTERRUPTS
9088 060516 000002 RTI ;; RETURN
9089 060520 005237 060322 32$: INC $STKCNT ;; COUNT THIS CHARACTER
9090 060524 021627 000140 CMP (SP),#140 ;; IS IT UPPER CASE?
9091 060530 002405 BLT 4$;; BRANCH IF YES
9092 060532 021627 000175 CMP (SP),#175 ;; IS IT A SPECIAL CHAR?
9093 060536 003002 BGT 4$;; BRANCH IF YES
9094 060540 042716 000040 BIC #40,(SP) ;; MAKE IT UPPER CASE
9095 060544 112677 177554 4$: MOVB (SP)+,$STKQIN ;; AND PUT IT IN QUEUE
9096 060550 005237 060324 INC $STKQIN ;; UPDATE THE POINTER
9097 060554 023727 060324 060331 CMP $STKQIN,$STKQEND ;; GO OFF THE END?
9098 060562 001003 BNE 5$;; BRANCH IF NO
9099 060564 012737 060330 060324 MOV #STKQSR,$STKQIN ;; RESET THE POINTER
9100 060572 000002 RTI ;; RETURN

```

```

9101
9102 ;; *****
9103 ;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
9104 ;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
9105 ;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
9106 ;; *CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

9107 060574 022737 000176 001140 $CKSWR: CMP #SWREG,$SWR ;; IS THE SOFT-SWR SELECTED
9108 060602 001104 BNE 15$;; EXIT IF NOT
9109 060604 105777 120334 TSTB $STKS ;; IS A CHAR WAITING?
9110 060610 100101 BPL 15$;; IF NOT, EXIT
9111 060612 117746 120330 MOVB $STKB,-(SP) ;; YES
9112 060616 042716 177600 BIC #177,(SP) ;; MAKE IT 7-BIT ASCII
9113 060622 021627 000007 CMP (SP),#7 ;; IS IT A CONTROL-G?
9114 060626 001300 BNE 2$;; IF NOT, PUT IT IN THE TTY QUEUE
9115 ;; AND EXIT

```

```

9116
9117 ;; *****
9118 ;; *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
9119 ;; *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
9120 ;; *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

9121 060630 123727 001134 000001 6$: CMPB $AUTOB,#1 ;; ARE WE RUNNING IN AUTO-MODE?
9122 060636 001674 BEQ 2$;; BRANCH IF YES
9123 060640 005726 TST (SP)+ ;; CLEAR CONTROL-G OFF STACK
9124 060642 004737 060332 JSR PC,$TKINT ;; FLUSH THE TTY INPUT QUEUE
9125 060646 005077 120272 CLR $STKS ;; DISABLE TTY KEYBOARD INTERRUPTS
9126 060652 112737 000001 001135 MOVB #1,$INTAG ;; SET INTERRUPT MODE INDICATOR
9127
9128 060660 104401 061436 SGT$WR: TYPE , $CNTLG ;; ECHO THE CONTROL-G (↑G)
9129 060664 104401 061443 TYPE , $MSWR ;; TYPE CURRENT CONTENTS
9130 060670 013746 000176 MOV $SWREG,-(SP) ;; SAVE $WREG FOR TYPEOUT
9131 060674 104402 TYPOC ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9132 060676 104401 061454 TYPE , $MNEW ;; PROMPT FOR NEW SWR
9133 060702 005046 19$: CLR -(SP) ;; CLEAR COUNTER
9134 060704 005046 CLR -(SP) ;; THE NEW SWR
9135 060706 105777 120232 7$: TSTB $STKS ;; CHAR THERE?
9136 060712 100375 BPL 7$;; IF NOT TRY AGAIN
9137
9138 060714 117746 120226 MOVB $STKB,-(SP) ;; PICK UP CHAR
9139 060720 042716 177600 BIC #177,(SP) ;; MAKE IT 7-BIT ASCII
9140

```





```

9197 061130 005337 060322 DEC STKCNT ;; DECREMENT THE COUNTER
9198 061134 117766 177166 000004 MOVB 2STKQOUT,4(SP) ;; GET ONE CHARACTER
9199 061142 005237 060326 INC STKQOUT ;; UPDATE THE POINTER
9200 061146 023727 060326 060331 CMP STKQOUT,#STKGEND ;; DID IT GO OFF OF THE END?
9201 061154 001003 BNE 2$;; BRANCH IF NO
9202 061156 012737 060330 060326 MOV #STKQSRST,STKQOUT ;; RESET THE POINTER
9203 061164 000002 RTI ;; RETURN
9204 ;; *****
9205 ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
9206 ;; CALL:
9207 ;; * RDLIN ;; INPUT A STRING FROM THE TTY
9208 ;; * RETURN HERE ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
9209 ;; * ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
9210
9211 061166 010346 $RDLIN: MOV R3,-(SP) ;; SAVE R3
9212 061170 005046 CLR -(SP) ;; CLEAR THE RUBOUT KEY
9213 061172 012703 061422 1$: MOV #STTYIN,R3 ;; GET ADDRESS
9214 061176 022703 061431 2$: CMP #STTYIN+7,R3 ;; BUFFER FULL?
9215 061202 101456 BLOS 4$;; BR IF YES
9216 061204 104410 RDCHR (SP)+,(R3) ;; GO READ ONE CHARACTER FROM THE TTY
9217 061206 112613 MOVB (R3),R3 ;; GET CHARACTER
9218 061210 122713 000177 10$: CMPB #177,(R3) ;; IS IT A RUBOUT
9219 061214 001022 BNE 5$;; BR IF NO
9220 061216 005716 TST (SP) ;; IS THIS THE FIRST RUBOUT?
9221 061220 001007 BNE 6$;; BR IF NO
9222 061222 112737 000134 061420 MOVB #' \,9$;; TYPE A BACK SLASH
9223 061230 104401 061420 TYPE 9$;;
9224 061234 012716 177777 MOV #-1,(SP) ;; SET THE RUBOUT KEY
9225 061240 005303 DEC R3 ;; BACKUP BY ONE
9226 061242 020327 061422 6$: CMP R3,#STTYIN ;; STACK EMPTY?
9227 061246 103434 BLO 4$;; BR IF YES
9228 061250 111337 061420 MOVB (R3),9$;; SETUP TO TYPEOUT THE DELETED CHAR.
9229 061254 104401 061420 TYPE 9$;; GO TYPE
9230 061260 000746 BR 2$;; GO READ ANOTHER CHAR.
9231 061262 005716 TST (SP) ;; RUBOUT KEY SET?
9232 061264 001406 BEQ 7$;; BR IF NO
9233 061266 112737 000134 061420 MOVB #' \,9$;; TYPE A BACK SLASH
9234 061274 104401 061420 TYPE 9$;;
9235 061300 005016 CLR (SP) ;; CLEAR THE RUBOUT KEY
9236 061302 122713 000025 7$: CMPB #25,(R3) ;; IS CHARACTER A CTRL U?
9237 061306 001003 BNE 8$;; BR IF NO
9238 061310 104401 061431 TYPE $CNTLU ;; TYPE A CONTROL "U"
9239 061314 000726 BR 1$;; GO START OVER
9240 061316 122713 000022 8$: CMPB #22,(R3) ;; IS CHARACTER A "R"?
9241 061322 001011 BNE 3$;; BRANCH IF NO
9242 061324 105013 CLRB (R3) ;; CLEAR THE CHARACTER
9243 061326 104401 001207 TYPE $SCRLF ;; TYPE A "CR" & "LF"
9244 061332 104401 061422 TYPE $STTYIN ;; TYPE THE INPUT STRING
9245 061336 000717 BR 2$;; GO PICKUP ANOTHER CHACTER
9246 061340 104401 001206 4$: TYPE $QUES ;; TYPE A '?'
9247 061344 000712 BR 1$;; CLEAR THE BUFFER AND LOOP
9248 061346 111337 061420 3$: MOVB (R3),9$;; ECHO THE CHARACTER
9249 061352 104401 061420 TYPE 9$;;
9250 061356 122723 000015 CMPB #15,(R3)+ ;; CHECK FOR RETURN
9251 061362 001305 BNE 2$;; LOOP IF NOT RETURN
9252 061364 105063 177777 CLRB -1(R3) ;; CLEAR RETURN (THE 15)

```

```

9253 061370 104401 001210 TYPE SLF ;; TYPE A LINE FEED
9254 061374 005726 TST (SP)+ ;; CLEAN RUBOUT KEY FROM THE STACK
9255 061376 012603 MOV (SP)+,R3 ;; RESTORE R3
9256 061400 011646 MOV (SP)-,(SP) ;; ADJUST THE STACK AND PUT ADDRESS OF THE
9257 061402 016666 000004 000002 MOV 4(SP),2(SP) ;; FIRST ASCII CHARACTER ON IT
9258 061410 012766 061422 000004 MOV #STTYIN,4(SP)
9259 061416 000002 RTI ;; RETURN
9260 061420 000 9$: .BYTE 0 ;; STORAGE FOR ASCII CHAR. TO TYPE
9261 061421 000 .BYTE 0 ;; TERMINATOR
9262 061422 000007 STTYIN: .BLKB 7 ;; RESERVE 7 BYTES FOR TTY INPUT
9263 061431 136 006525 000012 SCNTLU: .ASCIZ /↑U/<15><12> ;; CONTROL "U"
9264 061436 043536 005015 000 SCNTLG: .ASCIZ /↑G/<15><12> ;; CONTROL "G"
9265 061443 015 051412 051127 SMSWR: .ASCIZ <15><12>/SWR = /
9266 061450 036440 000040
9267 061454 020040 042516 020127 SMNEW: .ASCIZ / NEW = /
9268 061462 020075 000
9269 061466 .EVEN
9270 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
9271
9272 ;; *****
9273 ;; *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9274 ;; *CHANGE IT TO BINARY.
9275 ;; *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9276 ;; *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
9277 ;; *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9278 ;; *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9279 ;; *CALL:
9280 ;; * RDOCT ;; READ AN OCTAL NUMBER
9281 ;; * RETURN HERE ;; LOW ORDER BITS ARE ON TOP OF THE STACK
9282 ;; * ;; HIGH ORDER BITS ARE IN SHIOCT
9283
9284 061466 011646 SRDOCT: MOV (SP)-,(SP) ;; PROVIDE SPACE FOR THE
9285 061470 016666 000004 000002 MOV 4(SP),2(SP) ;; INPUT NUMBER
9286 061476 010046 MOV R0,-(SP) ;; PUSH R0 ON STACK
9287 061500 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
9288 061502 010246 MOV R2,-(SP) ;; PUSH R2 ON STACK
9289 061504 104411 1$: RDLIN ;; READ AN ASCII LINE
9290 061506 012600 MOV (SP)+,R0 ;; GET ADDRESS OF 1ST CHARACTER
9291 061510 010037 061614 MOV R0,5$;; AND SAVE IT
9292 061514 005001 CLR R1 ;; CLEAR DATA WORD
9293 061516 005002 CLR R2
9294 061520 112046 2$: MOVVB (R0)+,-(SP) ;; PICKUP THIS CHARACTER
9295 061522 001420 BEQ 3$;; IF ZERO GET OUT
9296 061524 122716 000060 CMPB #'0,(SP) ;; MAKE SURE THIS CHARACTER
9297 061530 003026 BGT 4$;; IS AN OCTAL DIGIT
9298 061532 122716 000067 CMPB #'7,(SP)
9299 061536 002423 BLT 4$
9300 061540 006301 ASL R1 ;; *2
9301 061542 006102 ROL R2
9302 061544 006301 ASL R1 ;; *4
9303 061546 006102 ROL R2
9304 061550 006301 ASL R1 ;; *8
9305 061552 006102 ROL R2
9306 061554 042716 177770 BIC #↑C7,(SP) ;; STRIP THE ASCII JUNK
9307 061560 062601 ADD (SP)+,R1 ;; ADD IN THIS DIGIT
9308 061562 000756 BR 2$;; LOOP

```

```

9309 061564 005726
9310 061566 010166 000012
9311 061572 010237 061624
9312 061576 012602
9313 061600 012601
9314 061602 012600
9315 061604 000002
9316 061606 005726
9317 061610 105010
9318 061612 104401
9319 061614 000000
9320 061616 104401 001206
9321 061622 000730
9322 061624 000000
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340 061626
9341 061626 010046
9342 061630 010146
9343 061632 010246
9344 061634 010346
9345 061636 010446
9346 061640 010546
9347 061642 016646 000022
9348 061646 016646 000022
9349 061652 016646 000022
9350 061656 016646 000022
9351 061662 000002
9352
9353
9354
9355
9356 061664
9357 061664 012666 000022
9358 061670 012666 000022
9359 061674 012666 000022
9360 061700 012666 000022
9361 061704 012605
9362 061706 012604
9363 061710 012603
9364 061712 012602

```

```

3S: TST (SP)+ ;: CLEAN TERMINATOR FROM STACK
 MOV R1,12(SP) ;: SAVE THE RESULT
 MOV R2,$SHIOCT
 MOV (SP)+,R2 ;: POP STACK INTO R2
 MOV (SP)+,R1 ;: POP STACK INTO R1
 MOV (SP)+,R0 ;: POP STACK INTO R0
 RTI ;: RETURN
4S: TST (SP)+ ;: CLEAN PARTIAL FROM STACK
 CLRB (R0) ;: SET A TERMINATOR
 TYPE ;: TYPE UP THRU THE BAD CHAR.
5S: .WORD 0
 TYPE $QUES ;: "?" "CR" & "LF"
 BR 1$;: TRY AGAIN
$SHIOCT: .WORD 0 ;: HIGH ORDER BITS GO HERE
.SBTTL SAVE AND RESTORE R0-R5 ROUTINES

```

```

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

```

```

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

```

```

;: *RESTORE R0-R5
;: *CALL:
;: * RESREG
$RESREG:
 MOV (SP)+,22(SP) ;: RESTORE PC OF CALL
 MOV (SP)+,22(SP) ;: RESTORE PS OF CALL
 MOV (SP)+,22(SP) ;: RESTORE PC OF MAIN FLOW
 MOV (SP)+,22(SP) ;: RESTORE PS OF MAIN FLOW
 MOV (SP)+,R5 ;: POP STACK INTO R5
 MOV (SP)+,R4 ;: POP STACK INTO R4
 MOV (SP)+,R3 ;: POP STACK INTO R3
 MOV (SP)+,R2 ;: POP STACK INTO R2

```

9365 061714 012601  
9366 061716 012600  
9367 061720 000002  
9368  
9369  
9370  
9371  
9372  
9373  
9374  
9375  
9376 061722 010046  
9377 061724 016600 000002  
9378 061730 005740  
9379 061732 111000  
9380 061734 006300  
9381 061736 016000 061756  
9382 061742 000200  
9383  
9384  
9385  
9386  
9387 061744 011646  
9388 061746 016666 000004 000002  
9389 061754 000002  
9390  
9391  
9392  
9393  
9394  
9395  
9396  
9397  
9398 061756 061744  
9399 061760 057430  
9400 061762 057674  
9401 061764 057650  
9402 061766 057710  
9403 061770 060076  
9404  
9405 061772 060664  
9406  
9407 061774 060574  
9408 061776 061076  
9409 062000 061166  
9410 062002 061466  
9411 062004 061626  
9412 062006 061664  
9413  
9414  
9415  
9416  
9417  
9418  
9419  
9420 062010 005015 055012 026532

```
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),R0 ;;GET TRAP ADDRESS
TST -(R0) ;;BACKUP BY 2
MOVB (R0),R0 ;;GET RIGHT BYTE OF TRAP
ASL R0 ;;POSITION FOR INDEXING
MOV $TRPAD(R0),R0 ;;INDEX TO TABLE
RTS R0 ;;GO TO ROUTINE
```

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

```
$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          |
|          | \$RDLIN  | ;;CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE             |
|          | \$RDOCT  | ;;CALL=RDOCT TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY         |
|          | \$SAVREG | ;;CALL=SAVREG TRAP+13(104413) SAVE RO-R5 ROUTINE                   |
|          | \$RESREG | ;;CALL=RESREG TRAP+14(104414) RESTORE RO-R5 ROUTINE                |

\*\*\*\*\*

.SBTTL TELETYPE MESSAGES

\*\*\*\*\*

TITLE: .ASCII <CR><LF><LF>/ZZ-CZRJE-B/<CR><LF>

|      |        |        |        |        |                                                                           |
|------|--------|--------|--------|--------|---------------------------------------------------------------------------|
| 9421 | 062016 | 055103 | 045122 | 026505 |                                                                           |
| 9422 | 062024 | 006502 | 012    |        |                                                                           |
| 9423 | 062027 | 122    | 030120 | 027464 | .ASCIZ BRP04/5/6 DUAL CONTROLLER LOGIC TEST - PART 12<CR><LF><LF>         |
| 9424 | 062034 | 027465 | 020066 | 052504 |                                                                           |
| 9425 | 062042 | 046101 | 041440 | 047117 |                                                                           |
| 9426 | 062050 | 051124 | 046117 | 042514 |                                                                           |
| 9427 | 062056 | 020122 | 047514 | 044507 |                                                                           |
| 9428 | 062064 | 020103 | 042524 | 052123 |                                                                           |
| 9429 | 062072 | 026440 | 050040 | 051101 |                                                                           |
| 9430 | 062100 | 020124 | 006461 | 005012 |                                                                           |
| 9431 | 062106 | 000    |        |        |                                                                           |
| 9432 | 062107 | 015    | 042412 | 052116 | ENTERA: .ASCIZ <CR><LF>/ENTER DRIVE ADDRESS: /                            |
| 9433 | 062114 | 051105 | 042040 | 044522 |                                                                           |
| 9434 | 062122 | 042526 | 040440 | 042104 |                                                                           |
| 9435 | 062130 | 042522 | 051523 | 020072 |                                                                           |
| 9436 | 062136 | 000    |        |        |                                                                           |
| 9437 | 062137 | 111    | 053116 | 046101 | ADRERR: .ASCIZ /INVALID ADDRESS/<CR><LF>                                  |
| 9438 | 062144 | 042111 | 040440 | 042104 |                                                                           |
| 9439 | 062152 | 042522 | 051523 | 005015 |                                                                           |
| 9440 | 062160 | 000    |        |        |                                                                           |
| 9441 | 062161 | 015    | 050012 | 051117 | PORTAIS: .ASCIZ <CR><LF>/PORT A ADDRESS IS: /                             |
| 9442 | 062166 | 020124 | 020101 | 042101 |                                                                           |
| 9443 | 062174 | 051104 | 051505 | 020123 |                                                                           |
| 9444 | 062202 | 051511 | 020072 | 000    | PORTBIS: .ASCIZ <CR><LF>/PORT B ADDRESS IS: /                             |
| 9445 | 062207 | 015    | 050012 | 051117 |                                                                           |
| 9446 | 062214 | 020124 | 020102 | 042101 |                                                                           |
| 9447 | 062222 | 051104 | 051505 | 020123 |                                                                           |
| 9448 | 062230 | 051511 | 020072 | 000    |                                                                           |
| 9449 | 062235 | 015    | 051412 | 051531 | NOCLOCK: .ASCIZ <CR><LF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CR><LF><LF>   |
| 9450 | 062242 | 042524 | 020115 | 052515 |                                                                           |
| 9451 | 062250 | 052123 | 044040 | 053101 |                                                                           |
| 9452 | 062256 | 020105 | 046047 | 020047 |                                                                           |
| 9453 | 062264 | 051117 | 023440 | 023520 |                                                                           |
| 9454 | 062272 | 041440 | 047514 | 045503 |                                                                           |
| 9455 | 062300 | 005015 | 000012 |        |                                                                           |
| 9456 | 062304 | 042412 | 052116 | 051105 | TESTNO: .ASCIZ <LF>/ENTER TEST #: /                                       |
| 9457 | 062312 | 052040 | 051505 | 020124 |                                                                           |
| 9458 | 062320 | 035043 | 000040 |        |                                                                           |
| 9459 | 062324 | 047111 | 040526 | 044514 | BADNO: .ASCIZ /INVALID TEST NUMBER/<CR><LF>                               |
| 9460 | 062332 | 020104 | 042524 | 052123 |                                                                           |
| 9461 | 062340 | 047040 | 046525 | 042502 |                                                                           |
| 9462 | 062346 | 006522 | 000012 |        |                                                                           |
| 9463 | 062352 | 042440 | 051122 | 051117 | TSTERR: .ASCIZ / ERRORS/<CR><LF>                                          |
| 9464 | 062360 | 006523 | 000012 |        |                                                                           |
| 9465 | 062364 | 005015 | 052012 | 042510 | ADDRIS: .ASCIZ <CR><LF><LF>/THE PRESENT ADDRESS OF THE RH11 (RPOS1) IS: / |
| 9466 | 062372 | 050040 | 042522 | 042523 |                                                                           |
| 9467 | 062400 | 052116 | 040440 | 042104 |                                                                           |
| 9468 | 062406 | 042522 | 051523 | 047440 |                                                                           |
| 9469 | 062414 | 020106 | 044124 | 020105 |                                                                           |
| 9470 | 062422 | 044122 | 030461 | 024040 |                                                                           |
| 9471 | 062430 | 050122 | 051503 | 024461 |                                                                           |
| 9472 | 062436 | 044440 | 035123 | 000040 |                                                                           |
| 9473 | 062444 | 042412 | 052116 | 051105 | NTRH11: .ASCIZ <LF>/ENTER NEW RH11 ADDRESS: /                             |
| 9474 | 062452 | 047040 | 053505 | 051040 |                                                                           |
| 9475 | 062460 | 030510 | 020061 | 042101 |                                                                           |
| 9476 | 062466 | 051104 | 051505 | 035123 |                                                                           |

```

9477 062474 000040
9478
9479
9480 ;:*****
9481 .SBTTL TEST ERROR MESSAGES
9482 ;:*****
9483
9484
9485 062476 051127 047117 020107 EM1: .ASCIZ /WRONG DRIVE TYPE/
9486 062504 051104 053111 020105
9487 062512 054524 042520 000
9488
9489 062517 104 044522 042526 EM2: .ASCIZ /DRIVE NOT ON LINE/
9490 062524 047040 052117 047440
9491 062532 020116 044514 042516
9492 062540 000
9493
9494 062541 123 051105 040511 EM3: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
9495 062546 020114 052516 041115
9496 062554 051105 051040 040505
9497 062562 020104 044124 047522
9498 062570 043525 020110 040505
9499 062576 044103 050040 051117
9500 062604 020124 047516 020124
9501 062612 044124 020105 040523
9502 062620 042515 000
9503
9504 062623 104 044522 042526 EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/
9505 062630 047040 052117 051440
9506 062636 044505 042532 020104
9507 062644 054502 050040 051117
9508 062652 000124
9509
9510 062654 051127 047117 020107 EM5: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
9511 062662 052123 052101 051525
9512 062670 051440 042505 020116
9513 062676 054502 052040 042510
9514 062704 051440 044505 044532
9515 062712 043516 050040 051117
9516 062720 000124
9517
9518 062722 042522 044507 052123 EM6: .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
9519 062730 051105 041440 047117
9520 062736 042524 052116 020123
9521 062744 042527 042522 051440
9522 062752 042505 020116 054502
9523 062760 047440 050120 051517
9524 062766 052111 020105 047520
9525 062774 052122 026440 042040
9526 063002 044522 042526 053440
9527 063010 051501 051440 044505
9528 063016 042532 000104
9529
9530 063022 042522 044507 052123 EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
9531 063030 051105 041440 047117
9532 063036 042524 052116 020123

```

|      |        |        |        |        |                                                                     |
|------|--------|--------|--------|--------|---------------------------------------------------------------------|
| 9533 | 063044 | 051127 | 047117 | 020107 |                                                                     |
| 9534 | 063052 | 043101 | 042524 | 020122 |                                                                     |
| 9535 | 063060 | 042522 | 042514 | 051501 |                                                                     |
| 9536 | 063066 | 020105 | 051117 | 052040 |                                                                     |
| 9537 | 063074 | 046511 | 047505 | 052125 |                                                                     |
| 9538 | 063102 | 000    |        |        |                                                                     |
| 9539 |        |        |        |        |                                                                     |
| 9540 | 063103 | 122    | 043505 | 051511 | EM10: .ASCIZ /REGISTER CONTENTS WRONG/                              |
| 9541 | 063110 | 042524 | 020122 | 047503 |                                                                     |
| 9542 | 063116 | 052116 | 047105 | 051524 |                                                                     |
| 9543 | 063124 | 053440 | 047522 | 043516 |                                                                     |
| 9544 | 063132 | 000    |        |        |                                                                     |
| 9545 |        |        |        |        |                                                                     |
| 9546 | 063133 | 103    | 047117 | 051124 | EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/  |
| 9547 | 063140 | 046117 | 041040 | 051525 |                                                                     |
| 9548 | 063146 | 050040 | 051101 | 052111 |                                                                     |
| 9549 | 063154 | 020131 | 051105 | 047522 |                                                                     |
| 9550 | 063162 | 020122 | 042522 | 042101 |                                                                     |
| 9551 | 063170 | 047111 | 020107 | 047111 |                                                                     |
| 9552 | 063176 | 044504 | 040503 | 042524 |                                                                     |
| 9553 | 063204 | 020104 | 042522 | 044507 |                                                                     |
| 9554 | 063212 | 052123 | 051105 | 000    |                                                                     |
| 9555 |        |        |        |        |                                                                     |
| 9556 | 063217 | 104    | 044522 | 042526 | EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/              |
| 9557 | 063224 | 047040 | 052117 | 051440 |                                                                     |
| 9558 | 063232 | 044505 | 042532 | 020104 |                                                                     |
| 9559 | 063240 | 054502 | 042040 | 044522 |                                                                     |
| 9560 | 063246 | 042526 | 041440 | 042514 |                                                                     |
| 9561 | 063254 | 051101 | 041440 | 046517 |                                                                     |
| 9562 | 063262 | 040515 | 042116 | 000    |                                                                     |
| 9563 |        |        |        |        |                                                                     |
| 9564 | 063267 | 122    | 040505 | 044504 | EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/ |
| 9565 | 063274 | 020116 | 051120 | 051505 |                                                                     |
| 9566 | 063302 | 052105 | 042040 | 042517 |                                                                     |
| 9567 | 063310 | 020123 | 047516 | 020124 |                                                                     |
| 9568 | 063316 | 042523 | 020124 | 047526 |                                                                     |
| 9569 | 063324 | 052514 | 042515 | 053040 |                                                                     |
| 9570 | 063332 | 046101 | 042111 | 043040 |                                                                     |
| 9571 | 063340 | 051117 | 052040 | 042510 |                                                                     |
| 9572 | 063346 | 050040 | 051117 | 000124 |                                                                     |
| 9573 |        |        |        |        |                                                                     |
| 9574 | 063354 | 047526 | 052514 | 042515 | EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/                   |
| 9575 | 063362 | 053040 | 046101 | 042111 |                                                                     |
| 9576 | 063370 | 051440 | 052105 | 047440 |                                                                     |
| 9577 | 063376 | 020116 | 044124 | 020105 |                                                                     |
| 9578 | 063404 | 051127 | 047117 | 020107 |                                                                     |
| 9579 | 063412 | 047520 | 052122 | 000    |                                                                     |
| 9580 |        |        |        |        |                                                                     |
| 9581 | 063417 | 101    | 052124 | 020116 | EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/       |
| 9582 | 063424 | 044502 | 020124 | 051127 |                                                                     |
| 9583 | 063432 | 047117 | 020107 | 043101 |                                                                     |
| 9584 | 063440 | 042524 | 020122 | 044524 |                                                                     |
| 9585 | 063446 | 042515 | 052517 | 020124 |                                                                     |
| 9586 | 063454 | 020055 | 042522 | 052521 |                                                                     |
| 9587 | 063462 | 051505 | 020124 | 047516 |                                                                     |
| 9588 | 063470 | 020124 | 042523 | 000124 |                                                                     |



|      |        |        |        |        |       |                                                               |
|------|--------|--------|--------|--------|-------|---------------------------------------------------------------|
| 9589 |        |        |        |        |       |                                                               |
| 9590 | 063476 | 052101 | 047124 | 041040 | EM16: | .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/           |
| 9591 | 063504 | 052111 | 053440 | 047522 |       |                                                               |
| 9592 | 063512 | 043516 | 040440 | 052106 |       |                                                               |
| 9593 | 063520 | 051105 | 051040 | 046105 |       |                                                               |
| 9594 | 063526 | 040505 | 042523 | 026440 |       |                                                               |
| 9595 | 063534 | 051040 | 050505 | 042525 |       |                                                               |
| 9596 | 063542 | 052123 | 051440 | 052105 |       |                                                               |
| 9597 | 063550 | 000    |        |        |       |                                                               |
| 9598 |        |        |        |        |       |                                                               |
| 9599 | 063551 | 101    | 052124 | 020116 | EM17: | .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/       |
| 9600 | 063556 | 044502 | 020124 | 051127 |       |                                                               |
| 9601 | 063564 | 047117 | 020107 | 043101 |       |                                                               |
| 9602 | 063572 | 042524 | 020122 | 042522 |       |                                                               |
| 9603 | 063600 | 042514 | 051501 | 020105 |       |                                                               |
| 9604 | 063606 | 020055 | 042522 | 052521 |       |                                                               |
| 9605 | 063614 | 051505 | 020124 | 047516 |       |                                                               |
| 9606 | 063622 | 020124 | 042523 | 000124 |       |                                                               |
| 9607 |        |        |        |        |       |                                                               |
| 9608 | 063630 | 051104 | 053111 | 020105 | EM20: | .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/      |
| 9609 | 063636 | 047516 | 020124 | 042523 |       |                                                               |
| 9610 | 063644 | 055111 | 042105 | 053440 |       |                                                               |
| 9611 | 063652 | 042510 | 020116 | 052101 |       |                                                               |
| 9612 | 063660 | 047124 | 041040 | 052111 |       |                                                               |
| 9613 | 063666 | 043040 | 051117 | 050040 |       |                                                               |
| 9614 | 063674 | 051117 | 020124 | 046103 |       |                                                               |
| 9615 | 063702 | 040505 | 042522 | 000104 |       |                                                               |
| 9616 |        |        |        |        |       |                                                               |
| 9617 | 063710 | 051104 | 053111 | 020105 | EM21: | .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/           |
| 9618 | 063716 | 042523 | 055111 | 042105 |       |                                                               |
| 9619 | 063724 | 053440 | 042510 | 020116 |       |                                                               |
| 9620 | 063732 | 042532 | 047522 | 053440 |       |                                                               |
| 9621 | 063740 | 044522 | 052124 | 047105 |       |                                                               |
| 9622 | 063746 | 044440 | 020116 | 052101 |       |                                                               |
| 9623 | 063754 | 047124 | 041040 | 052111 |       |                                                               |
| 9624 | 063762 | 000    |        |        |       |                                                               |
| 9625 |        |        |        |        |       |                                                               |
| 9626 | 063763 | 104    | 044522 | 042526 | EM22: | .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/ |
| 9627 | 063770 | 047040 | 052117 | 044440 |       |                                                               |
| 9628 | 063776 | 020116 | 042516 | 052125 |       |                                                               |
| 9629 | 064004 | 040522 | 020114 | 043101 |       |                                                               |
| 9630 | 064012 | 042524 | 020122 | 044524 |       |                                                               |
| 9631 | 064020 | 042515 | 052517 | 020124 |       |                                                               |
| 9632 | 064026 | 020055 | 042522 | 052521 |       |                                                               |
| 9633 | 064034 | 051505 | 020124 | 047516 |       |                                                               |
| 9634 | 064042 | 020124 | 042523 | 000124 |       |                                                               |
| 9635 |        |        |        |        |       |                                                               |
| 9636 | 064050 | 044524 | 042515 | 052517 | EM23: | .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/                |
| 9637 | 064056 | 020124 | 046103 | 040505 |       |                                                               |
| 9638 | 064064 | 042522 | 020104 | 044124 |       |                                                               |
| 9639 | 064072 | 020105 | 051104 | 053111 |       |                                                               |
| 9640 | 064100 | 023505 | 020123 | 051105 |       |                                                               |
| 9641 | 064106 | 047522 | 020122 | 044502 |       |                                                               |
| 9642 | 064114 | 000124 |        |        |       |                                                               |
| 9643 |        |        |        |        |       |                                                               |
| 9644 | 064116 | 042522 | 042514 | 051501 | EM24: | .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/       |

|      |        |        |        |        |                                                                             |
|------|--------|--------|--------|--------|-----------------------------------------------------------------------------|
| 9645 | 064124 | 020105 | 047503 | 046515 |                                                                             |
| 9646 | 064132 | 047101 | 020104 | 042522 |                                                                             |
| 9647 | 064140 | 042514 | 051501 | 042105 |                                                                             |
| 9648 | 064146 | 042040 | 044522 | 042526 |                                                                             |
| 9649 | 064154 | 053440 | 052111 | 020110 |                                                                             |
| 9650 | 064162 | 051105 | 047522 | 051522 |                                                                             |
| 9651 | 064170 | 051440 | 052105 | 000    |                                                                             |
| 9652 |        |        |        |        |                                                                             |
| 9653 | 064175 | 124    | 046511 | 047505 | EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/                           |
| 9654 | 064202 | 052125 | 047440 | 042516 |                                                                             |
| 9655 | 064210 | 051455 | 047510 | 020124 |                                                                             |
| 9656 | 064216 | 044504 | 020104 | 047516 |                                                                             |
| 9657 | 064224 | 020124 | 042522 | 051124 |                                                                             |
| 9658 | 064232 | 043511 | 042507 | 000122 |                                                                             |
| 9659 |        |        |        |        |                                                                             |
| 9660 | 064240 | 051104 | 053111 | 020105 | EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/         |
| 9661 | 064246 | 047516 | 020124 | 047111 |                                                                             |
| 9662 | 064254 | 047040 | 052505 | 051124 |                                                                             |
| 9663 | 064262 | 046101 | 040440 | 052106 |                                                                             |
| 9664 | 064270 | 051105 | 051040 | 046105 |                                                                             |
| 9665 | 064276 | 040505 | 042523 | 026440 |                                                                             |
| 9666 | 064304 | 051040 | 050505 | 042525 |                                                                             |
| 9667 | 064312 | 052123 | 047040 | 052117 |                                                                             |
| 9668 | 064320 | 051440 | 052105 | 000    |                                                                             |
| 9669 |        |        |        |        |                                                                             |
| 9670 | 064325 | 122    | 043505 | 051511 | EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/                |
| 9671 | 064332 | 042524 | 020122 | 051127 |                                                                             |
| 9672 | 064340 | 047117 | 020107 | 043101 |                                                                             |
| 9673 | 064346 | 042524 | 020122 | 042522 |                                                                             |
| 9674 | 064354 | 042514 | 051501 | 020105 |                                                                             |
| 9675 | 064362 | 044527 | 044124 | 051040 |                                                                             |
| 9676 | 064370 | 050505 | 042525 | 052123 |                                                                             |
| 9677 | 064376 | 051440 | 052105 | 000    |                                                                             |
| 9678 |        |        |        |        |                                                                             |
| 9679 | 064403 | 104    | 044522 | 042526 | EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/ |
| 9680 | 064410 | 051440 | 044505 | 042532 |                                                                             |
| 9681 | 064416 | 020104 | 054502 | 051040 |                                                                             |
| 9682 | 064424 | 046105 | 040505 | 042523 |                                                                             |
| 9683 | 064432 | 041440 | 046517 | 040515 |                                                                             |
| 9684 | 064440 | 042116 | 044440 | 051523 |                                                                             |
| 9685 | 064446 | 042525 | 020104 | 044127 |                                                                             |
| 9686 | 064454 | 047105 | 042040 | 044522 |                                                                             |
| 9687 | 064462 | 042526 | 044440 | 020116 |                                                                             |
| 9688 | 064470 | 042516 | 052125 | 040522 |                                                                             |
| 9689 | 064476 | 000114 |        |        |                                                                             |
| 9690 |        |        |        |        |                                                                             |
| 9691 | 064500 | 051104 | 053111 | 020105 | EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/                 |
| 9692 | 064506 | 047111 | 047040 | 052505 |                                                                             |
| 9693 | 064514 | 051124 | 046101 | 040440 |                                                                             |
| 9694 | 064522 | 052106 | 051105 | 051040 |                                                                             |
| 9695 | 064530 | 046105 | 040505 | 042523 |                                                                             |
| 9696 | 064536 | 026440 | 051040 | 050505 |                                                                             |
| 9697 | 064544 | 042525 | 052123 | 051440 |                                                                             |
| 9698 | 064552 | 052105 | 000    |        |                                                                             |
| 9699 |        |        |        |        |                                                                             |
| 9700 | 064555 | 101    | 052124 | 020116 | EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/                     |

|      |        |        |        |        |
|------|--------|--------|--------|--------|
| 9701 | 064562 | 044502 | 020124 | 051127 |
| 9702 | 064570 | 047117 | 020107 | 043101 |
| 9703 | 064576 | 042524 | 020122 | 042522 |
| 9704 | 064604 | 040503 | 044514 | 051102 |
| 9705 | 064612 | 052101 | 020105 | 047503 |
| 9706 | 064620 | 046515 | 047101 | 000104 |
| 9707 |        |        |        |        |
| 9708 | 064626 | 051104 | 053111 | 020105 |
| 9709 | 064634 | 042522 | 052524 | 047122 |
| 9710 | 064642 | 042105 | 052040 | 020117 |
| 9711 | 064650 | 042516 | 052125 | 040522 |
| 9712 | 064656 | 020114 | 043111 | 042040 |
| 9713 | 064664 | 044522 | 042526 | 041440 |
| 9714 | 064672 | 042514 | 051101 | 043440 |
| 9715 | 064700 | 053111 | 047105 | 053440 |
| 9716 | 064706 | 044510 | 042514 | 042040 |
| 9717 | 064714 | 044522 | 042526 | 051440 |
| 9718 | 064722 | 044505 | 042532 | 000104 |
| 9719 |        |        |        |        |
| 9720 | 064730 | 051104 | 053111 | 020105 |
| 9721 | 064736 | 042522 | 052524 | 047122 |
| 9722 | 064744 | 042105 | 052040 | 020117 |
| 9723 | 064752 | 042516 | 052125 | 040522 |
| 9724 | 064760 | 020114 | 043111 | 046440 |
| 9725 | 064766 | 051501 | 041123 | 051525 |
| 9726 | 064774 | 044440 | 044516 | 020124 |
| 9727 | 065002 | 044507 | 042526 | 020116 |
| 9728 | 065010 | 044127 | 046111 | 020105 |
| 9729 | 065016 | 051104 | 053111 | 020105 |
| 9730 | 065024 | 042523 | 055111 | 042105 |
| 9731 | 065032 | 000    |        |        |
| 9732 |        |        |        |        |

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

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

|      |        |        |        |        |                                                                      |
|------|--------|--------|--------|--------|----------------------------------------------------------------------|
| 9733 | 065033 | 124    | 046511 | 047505 | EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/        |
| 9734 | 065040 | 052125 | 047440 | 042516 |                                                                      |
| 9735 | 065046 | 051440 | 047510 | 020124 |                                                                      |
| 9736 | 065054 | 044506 | 042522 | 020104 |                                                                      |
| 9737 | 065062 | 044527 | 044124 | 052517 |                                                                      |
| 9738 | 065070 | 020124 | 042522 | 044507 |                                                                      |
| 9739 | 065076 | 052123 | 051105 | 040440 |                                                                      |
| 9740 | 065104 | 041503 | 051505 | 000123 |                                                                      |
| 9741 |        |        |        |        |                                                                      |
| 9742 | 065112 | 044524 | 042515 | 052517 | EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/             |
| 9743 | 065120 | 020124 | 040510 | 020123 |                                                                      |
| 9744 | 065126 | 047516 | 020124 | 041517 |                                                                      |
| 9745 | 065134 | 052503 | 051122 | 042105 |                                                                      |
| 9746 | 065142 | 053440 | 052111 | 044510 |                                                                      |
| 9747 | 065150 | 020116 | 020062 | 042523 |                                                                      |
| 9748 | 065156 | 047503 | 042116 | 000123 |                                                                      |
| 9749 |        |        |        |        |                                                                      |
| 9750 | 065164 | 051104 | 053111 | 020105 | EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/                 |
| 9751 | 065172 | 051511 | 047040 | 047117 |                                                                      |
| 9752 | 065200 | 042455 | 044530 | 052123 |                                                                      |
| 9753 | 065206 | 047105 | 020124 | 023450 |                                                                      |
| 9754 | 065214 | 042516 | 023504 | 041040 |                                                                      |
| 9755 | 065222 | 052111 | 051440 | 052105 |                                                                      |
| 9756 | 065230 | 000051 |        |        |                                                                      |
| 9757 |        |        |        |        |                                                                      |
| 9758 | 065232 | 052101 | 047124 | 041040 | EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/          |
| 9759 | 065240 | 052111 | 043040 | 051117 |                                                                      |
| 9760 | 065246 | 050040 | 051117 | 020124 |                                                                      |
| 9761 | 065254 | 047516 | 020124 | 042522 |                                                                      |
| 9762 | 065262 | 042523 | 020124 | 054502 |                                                                      |
| 9763 | 065270 | 046440 | 051501 | 041123 |                                                                      |
| 9764 | 065276 | 051525 | 041440 | 042514 |                                                                      |
| 9765 | 065304 | 051101 | 000    |        |                                                                      |
| 9766 |        |        |        |        |                                                                      |
| 9767 | 065307 | 124    | 046511 | 047505 | EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/                     |
| 9768 | 065314 | 052125 | 041440 | 042514 |                                                                      |
| 9769 | 065322 | 051101 | 042105 | 052040 |                                                                      |
| 9770 | 065330 | 042510 | 040440 | 052124 |                                                                      |
| 9771 | 065336 | 047105 | 044524 | 047117 |                                                                      |
| 9772 | 065344 | 041040 | 052111 | 000    |                                                                      |
| 9773 |        |        |        |        |                                                                      |
| 9774 | 065351 | 104    | 044522 | 042526 | EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/ |
| 9775 | 065356 | 047040 | 052117 | 044440 |                                                                      |
| 9776 | 065364 | 020116 | 042516 | 052125 |                                                                      |
| 9777 | 065372 | 040522 | 020114 | 051117 |                                                                      |
| 9778 | 065400 | 051440 | 044505 | 042532 |                                                                      |
| 9779 | 065406 | 020104 | 043101 | 042524 |                                                                      |
| 9780 | 065414 | 020122 | 052101 | 047124 |                                                                      |
| 9781 | 065422 | 041040 | 052111 | 053440 |                                                                      |
| 9782 | 065430 | 044522 | 052124 | 047105 |                                                                      |
| 9783 | 065436 | 000    |        |        |                                                                      |
| 9784 |        |        |        |        |                                                                      |
| 9785 | 065437 | 104    | 044522 | 042526 | EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/          |
| 9786 | 065444 | 044440 | 020116 | 042516 |                                                                      |
| 9787 | 065452 | 052125 | 040522 | 020114 |                                                                      |
| 9788 | 065460 | 043101 | 042524 | 020122 |                                                                      |

|      |        |        |        |        |                                                                      |
|------|--------|--------|--------|--------|----------------------------------------------------------------------|
| 9789 | 065466 | 052101 | 042524 | 052116 |                                                                      |
| 9790 | 065474 | 047511 | 020116 | 044502 |                                                                      |
| 9791 | 065502 | 020124 | 051127 | 052111 |                                                                      |
| 9792 | 065510 | 042524 | 000116 |        |                                                                      |
| 9793 |        |        |        |        |                                                                      |
| 9794 | 065514 | 051127 | 052111 | 020105 | EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/          |
| 9795 | 065522 | 052101 | 042524 | 052116 |                                                                      |
| 9796 | 065530 | 047511 | 020116 | 044502 |                                                                      |
| 9797 | 065536 | 020124 | 044504 | 020104 |                                                                      |
| 9798 | 065544 | 047516 | 020124 | 042523 |                                                                      |
| 9799 | 065552 | 020124 | 047520 | 052122 |                                                                      |
| 9800 | 065560 | 051040 | 050505 | 042525 |                                                                      |
| 9801 | 065566 | 052123 | 000    |        |                                                                      |
| 9802 |        |        |        |        |                                                                      |
| 9803 | 065571 | 103    | 047117 | 051124 | EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@        |
| 9804 | 065576 | 046117 | 042514 | 020122 |                                                                      |
| 9805 | 065604 | 042523 | 042514 | 052103 |                                                                      |
| 9806 | 065612 | 051440 | 044527 | 041524 |                                                                      |
| 9807 | 065620 | 020110 | 047117 | 042040 |                                                                      |
| 9808 | 065626 | 044522 | 042526 | 047040 |                                                                      |
| 9809 | 065634 | 052117 | 044440 | 020116 |                                                                      |
| 9810 | 065642 | 040447 | 041057 | 000047 |                                                                      |
| 9811 |        |        |        |        |                                                                      |
| 9812 | 065650 | 040503 | 023516 | 020124 | EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/                |
| 9813 | 065656 | 041501 | 042503 | 051523 |                                                                      |
| 9814 | 065664 | 042040 | 044522 | 042526 |                                                                      |
| 9815 | 065672 | 052040 | 051110 | 052517 |                                                                      |
| 9816 | 065700 | 044107 | 042440 | 052111 |                                                                      |
| 9817 | 065706 | 042510 | 020122 | 047520 |                                                                      |
| 9818 | 065714 | 052122 | 000    |        |                                                                      |
| 9819 |        |        |        |        |                                                                      |
| 9820 | 065717 | 101    | 052124 | 020116 | EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/ |
| 9821 | 065724 | 044502 | 020124 | 047506 |                                                                      |
| 9822 | 065732 | 020122 | 042523 | 055111 |                                                                      |
| 9823 | 065740 | 047111 | 020107 | 047520 |                                                                      |
| 9824 | 065746 | 052122 | 047040 | 052117 |                                                                      |
| 9825 | 065754 | 041440 | 042514 | 051101 |                                                                      |
| 9826 | 065762 | 042105 | 041040 | 020131 |                                                                      |
| 9827 | 065770 | 040515 | 051523 | 052502 |                                                                      |
| 9828 | 065776 | 020123 | 047111 | 052111 |                                                                      |
| 9829 | 066004 | 000    |        |        |                                                                      |
| 9830 |        |        |        |        |                                                                      |
| 9831 | 066005 | 101    | 052124 | 020116 | EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/    |
| 9832 | 066012 | 044502 | 020124 | 047506 |                                                                      |
| 9833 | 066020 | 020122 | 050117 | 047520 |                                                                      |
| 9834 | 066026 | 044523 | 042524 | 050040 |                                                                      |
| 9835 | 066034 | 051117 | 020124 | 046103 |                                                                      |
| 9836 | 066042 | 040505 | 042522 | 020104 |                                                                      |
| 9837 | 066050 | 054502 | 046440 | 051501 |                                                                      |
| 9838 | 066056 | 041123 | 051525 | 044440 |                                                                      |
| 9839 | 066064 | 044516 | 000124 |        |                                                                      |
| 9840 |        |        |        |        |                                                                      |
| 9841 | 066070 | 052101 | 047124 | 041040 | EM51: .ASCIZ /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/    |
| 9842 | 066076 | 052111 | 041440 | 042514 |                                                                      |
| 9843 | 066104 | 051101 | 042105 | 041040 |                                                                      |
| 9844 | 066112 | 020131 | 040515 | 051523 |                                                                      |

9845 066120 052502 020123 047111  
 9846 066126 052111 020054 051104  
 9847 066134 053111 020105 047111  
 9848 066142 047040 052505 051124  
 9849 066150 046101 000  
 9850  
 9851 066153 124 042510 040440  
 9852 066160 052124 020116 044502  
 9853 066166 020124 051511 051440  
 9854 066174 052105 040440 052106  
 9855 066202 051105 052040 046511  
 9856 066210 047505 052125 053440  
 9857 066216 052111 020110 047516  
 9858 066224 051040 050505 042525  
 9859 066232 052123 023040 023440  
 9860 066240 051105 023522 051440  
 9861 066246 052105 000  
 9862  
 9863 066251 103 047101 052047  
 9864 066256 051040 040505 020104  
 9865 066264 044124 020105 052101  
 9866 066272 047124 041040 052111  
 9867 066300 043040 047522 020115  
 9868 066306 044124 020105 047447  
 9869 066314 050120 051517 052111  
 9870 066322 023505 050040 051117  
 9871 066330 000124  
 9872  
 9873 066332 042522 042514 051501  
 9874 066340 020105 047503 046515  
 9875 066346 047101 020104 042522  
 9876 066354 047503 047107 055111  
 9877 066362 042105 053440 042510  
 9878 066370 020116 051511 052523  
 9879 066376 042105 041040 020131  
 9880 066404 047516 026516 042523  
 9881 066412 055111 047111 020107  
 9882 066420 047520 052122 000  
 9883  
 9884 066425 124 046511 047505  
 9885 066432 052125 047440 042516  
 9886 066440 051455 047510 020124  
 9887 066446 051511 046040 051505  
 9888 066454 020123 044124 047101  
 9889 066462 032440 030060 046440  
 9890 066470 000123  
 9891  
 9892 066472 044122 030461 042040  
 9893 066500 042111 023516 020124  
 9894 066506 042522 050123 047117  
 9895 066514 020104 047524 040440  
 9896 066522 042104 042522 051523  
 9897 066530 047111 000107  
 9898  
 9899  
 9900

EMS2: .ASCIZ /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/

EMS3: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/

EMS4: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/

EMS5: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/

EMS6: .ASCIZ /RH11 DIDN'T RESPOND TO ADDRESSING/



|       |        |        |        |        |       |                |        |                 |                 |           |           |
|-------|--------|--------|--------|--------|-------|----------------|--------|-----------------|-----------------|-----------|-----------|
| 9957  | 067212 | 041520 | 020040 | 047520 |       |                |        |                 |                 |           |           |
| 9958  | 067220 | 052122 | 021440 | 020040 |       |                |        |                 |                 |           |           |
| 9959  | 067226 | 042522 | 020107 | 042101 |       |                |        |                 |                 |           |           |
| 9960  | 067234 | 020122 | 047503 | 052116 |       |                |        |                 |                 |           |           |
| 9961  | 067242 | 047105 | 051524 | 000    |       |                |        |                 |                 |           |           |
| 9962  | 067247 | 040    | 020040 | 020040 | DH13: | .ASCII /       |        | SEIZE           | ERROR/<CR><LF>  |           |           |
| 9963  | 067254 | 020040 | 020040 | 020040 |       |                |        |                 |                 |           |           |
| 9964  | 067262 | 020040 | 020040 | 051440 |       |                |        |                 |                 |           |           |
| 9965  | 067270 | 044505 | 042532 | 020040 |       |                |        |                 |                 |           |           |
| 9966  | 067276 | 042440 | 051122 | 051117 |       |                |        |                 |                 |           |           |
| 9967  | 067304 | 005015 |        |        |       |                |        |                 |                 |           |           |
| 9968  | 067306 | 042524 | 052123 | 021440 |       | .ASCIZ /TEST # | ERR PC | PORT #          | PORT #          | REG ADR   | CONTENTS/ |
| 9969  | 067314 | 020040 | 051105 | 020122 |       |                |        |                 |                 |           |           |
| 9970  | 067322 | 041520 | 020040 | 047520 |       |                |        |                 |                 |           |           |
| 9971  | 067330 | 052122 | 021440 | 020040 |       |                |        |                 |                 |           |           |
| 9972  | 067336 | 047520 | 052122 | 021440 |       |                |        |                 |                 |           |           |
| 9973  | 067344 | 020040 | 042522 | 020107 |       |                |        |                 |                 |           |           |
| 9974  | 067352 | 042101 | 020122 | 047503 |       |                |        |                 |                 |           |           |
| 9975  | 067360 | 052116 | 047105 | 051524 |       |                |        |                 |                 |           |           |
| 9976  | 067366 | 000    |        |        |       |                |        |                 |                 |           |           |
| 9977  | 067367 | 040    | 020040 | 020040 | DH22: | .ASCII /       |        | RELSNG          | SEIZE/<CR><LF>  |           |           |
| 9978  | 067374 | 020040 | 020040 | 020040 |       |                |        |                 |                 |           |           |
| 9979  | 067402 | 020040 | 020040 | 051040 |       |                |        |                 |                 |           |           |
| 9980  | 067410 | 046105 | 047123 | 020107 |       |                |        |                 |                 |           |           |
| 9981  | 067416 | 051440 | 044505 | 042532 |       |                |        |                 |                 |           |           |
| 9982  | 067424 | 005015 |        |        |       |                |        |                 |                 |           |           |
| 9983  | 067426 | 042524 | 052123 | 021440 |       | .ASCIZ /TEST # | ERR PC | PORT #          | PORT #/         |           |           |
| 9984  | 067434 | 020040 | 051105 | 020122 |       |                |        |                 |                 |           |           |
| 9985  | 067442 | 041520 | 020040 | 047520 |       |                |        |                 |                 |           |           |
| 9986  | 067450 | 052122 | 021440 | 020040 |       |                |        |                 |                 |           |           |
| 9987  | 067456 | 047520 | 052122 | 021440 |       |                |        |                 |                 |           |           |
| 9988  | 067464 | 000    |        |        |       |                |        |                 |                 |           |           |
| 9989  | 067465 | 040    | 020040 | 020040 | DH23: | .ASCII /       |        | SEIZE/<CR><LF>  |                 |           |           |
| 9990  | 067472 | 020040 | 020040 | 020040 |       |                |        |                 |                 |           |           |
| 9991  | 067500 | 020040 | 020040 | 051440 |       |                |        |                 |                 |           |           |
| 9992  | 067506 | 044505 | 042532 | 005015 |       |                |        |                 |                 |           |           |
| 9993  | 067514 | 042524 | 052123 | 021440 |       | .ASCIZ /TEST # | ERR PC | PORT #          | REG ADR         | CONTENTS/ |           |
| 9994  | 067522 | 020040 | 051105 | 020122 |       |                |        |                 |                 |           |           |
| 9995  | 067530 | 041520 | 020040 | 047520 |       |                |        |                 |                 |           |           |
| 9996  | 067536 | 052122 | 021440 | 020040 |       |                |        |                 |                 |           |           |
| 9997  | 067544 | 042522 | 020107 | 042101 |       |                |        |                 |                 |           |           |
| 9998  | 067552 | 020122 | 047503 | 052116 |       |                |        |                 |                 |           |           |
| 9999  | 067560 | 047105 | 051524 | 000    |       |                |        |                 |                 |           |           |
| 10000 | 067565 | 040    | 020040 | 020040 | DH26: | .ASCII /       |        | RELSNG/<CR><LF> |                 |           |           |
| 10001 | 067572 | 020040 | 020040 | 020040 |       |                |        |                 |                 |           |           |
| 10002 | 067600 | 020040 | 020040 | 051040 |       |                |        |                 |                 |           |           |
| 10003 | 067606 | 046105 | 047123 | 006507 |       |                |        |                 |                 |           |           |
| 10004 | 067614 | 012    |        |        |       |                |        |                 |                 |           |           |
| 10005 | 067615 | 124    | 051505 | 020124 |       | .ASCIZ /TEST # | ERR PC | PORT #/         |                 |           |           |
| 10006 | 067622 | 020043 | 042440 | 051122 |       |                |        |                 |                 |           |           |
| 10007 | 067630 | 050040 | 020103 | 050040 |       |                |        |                 |                 |           |           |
| 10008 | 067636 | 051117 | 020124 | 000043 |       |                |        |                 |                 |           |           |
| 10009 | 067644 | 020040 | 020040 | 020040 | DH31: | .ASCII /       |        | RELSNG          | RQSTNG/<CR><LF> |           |           |
| 10010 | 067652 | 020040 | 020040 | 020040 |       |                |        |                 |                 |           |           |
| 10011 | 067660 | 020040 | 020040 | 042522 |       |                |        |                 |                 |           |           |
| 10012 | 067666 | 051514 | 043516 | 020040 |       |                |        |                 |                 |           |           |



|       |        |        |        |        |       |        |                                               |         |                               |
|-------|--------|--------|--------|--------|-------|--------|-----------------------------------------------|---------|-------------------------------|
| 10013 | 067674 | 050522 | 052123 | 043516 |       |        |                                               |         |                               |
| 10014 | 067702 | 005015 |        |        |       |        |                                               |         |                               |
| 10015 | 067704 | 042524 | 052123 | 021440 |       | .ASCIZ | /TEST                                         | ERR PC  | PORT # PORT #/                |
| 10016 | 067712 | 020040 | 051105 | 020122 |       |        |                                               |         |                               |
| 10017 | 067720 | 041520 | 020040 | 047520 |       |        |                                               |         |                               |
| 10018 | 067726 | 052122 | 021440 | 020040 |       |        |                                               |         |                               |
| 10019 | 067734 | 047520 | 052122 | 021440 |       |        |                                               |         |                               |
| 10020 | 067742 | 000    |        |        |       |        |                                               |         |                               |
| 10021 | 067743 | 124    | 051505 | 020124 | DH36: | .ASCIZ | /TEST                                         | ERR PC  | PORT #/                       |
| 10022 | 067750 | 020043 | 042440 | 051122 |       |        |                                               |         |                               |
| 10023 | 067756 | 050040 | 020103 | 050040 |       |        |                                               |         |                               |
| 10024 | 067764 | 051117 | 020124 | 000043 |       |        |                                               |         |                               |
| 10025 | 067772 | 042524 | 052123 | 021440 | DH42: | .ASCIZ | /TEST                                         | ERR PC/ |                               |
| 10026 | 070000 | 020040 | 051105 | 020122 |       |        |                                               |         |                               |
| 10027 | 070006 | 041520 | 000    |        |       |        |                                               |         |                               |
| 10028 | 070011 | 040    | 020040 | 020040 | DH44: | .ASCII | /                                             |         | RELSNG ERROR/<CR><LF>         |
| 10029 | 070016 | 020040 | 020040 | 020040 |       |        |                                               |         |                               |
| 10030 | 070024 | 020040 | 020040 | 051040 |       |        |                                               |         |                               |
| 10031 | 070032 | 046105 | 047123 | 020107 |       |        |                                               |         |                               |
| 10032 | 070040 | 042440 | 051122 | 051117 |       |        |                                               |         |                               |
| 10033 | 070046 | 005015 |        |        |       |        |                                               |         |                               |
| 10034 | 070050 | 042524 | 052123 | 021440 |       | .ASCIZ | /TEST                                         | ERR PC  | PORT # PORT #/                |
| 10035 | 070056 | 020040 | 051105 | 020122 |       |        |                                               |         |                               |
| 10036 | 070064 | 041520 | 020040 | 047520 |       |        |                                               |         |                               |
| 10037 | 070072 | 052122 | 021440 | 020040 |       |        |                                               |         |                               |
| 10038 | 070100 | 047520 | 052122 | 021440 |       |        |                                               |         |                               |
| 10039 | 070106 | 000    |        |        |       |        |                                               |         |                               |
| 10040 | 070107 | 040    | 020040 | 020040 | DH46: | .ASCII | /                                             |         | PORT A PORT B/<CR><LF>        |
| 10041 | 070114 | 020040 | 020040 | 020040 |       |        |                                               |         |                               |
| 10042 | 070122 | 020040 | 020040 | 050040 |       |        |                                               |         |                               |
| 10043 | 070130 | 051117 | 020124 | 020101 |       |        |                                               |         |                               |
| 10044 | 070136 | 050040 | 051117 | 020124 |       |        |                                               |         |                               |
| 10045 | 070144 | 006502 | 012    |        |       |        |                                               |         |                               |
| 10046 | 070147 | 124    | 051505 | 020124 |       | .ASCIZ | /TEST                                         | ERR PC  | RPDS1 RPDS1/                  |
| 10047 | 070154 | 020043 | 042440 | 051122 |       |        |                                               |         |                               |
| 10048 | 070162 | 050040 | 020103 | 051040 |       |        |                                               |         |                               |
| 10049 | 070170 | 042120 | 030523 | 020040 |       |        |                                               |         |                               |
| 10050 | 070176 | 051040 | 042120 | 030523 |       |        |                                               |         |                               |
| 10051 | 070204 | 000    |        |        |       |        |                                               |         |                               |
| 10052 | 070205 | 124    | 051505 | 020124 | DH55: | .ASCIZ | /TEST                                         | ERR PC  | PORT # TIMEOUT VALUE (IN MS)/ |
| 10053 | 070212 | 020043 | 042440 | 051122 |       |        |                                               |         |                               |
| 10054 | 070220 | 050040 | 020103 | 050040 |       |        |                                               |         |                               |
| 10055 | 070226 | 051117 | 020124 | 020043 |       |        |                                               |         |                               |
| 10056 | 070234 | 052040 | 046511 | 047505 |       |        |                                               |         |                               |
| 10057 | 070242 | 052125 | 053040 | 046101 |       |        |                                               |         |                               |
| 10058 | 070250 | 042525 | 024040 | 047111 |       |        |                                               |         |                               |
| 10059 | 070256 | 046440 | 024523 | 000    |       |        |                                               |         |                               |
| 10060 | 070263 | 044    | 050122 | 042101 | DH56: | .ASCIZ | /SRPADR/                                      |         |                               |
| 10061 | 070270 | 000122 |        |        |       |        |                                               |         |                               |
| 10062 |        |        |        |        |       |        |                                               |         |                               |
| 10063 |        |        |        |        |       | .EVEN  |                                               |         |                               |
| 10064 |        |        |        |        |       |        |                                               |         |                               |
| 10065 | 070272 | 001242 | 001116 | 001234 | DT1:  | .WORD  | TSTNUM, \$ERRPC, PTNBR, \$BDADR, \$BDDAT, 0   |         |                               |
| 10066 | 070300 | 001122 | 001126 | 000000 |       |        |                                               |         |                               |
| 10067 | 070306 | 001242 | 001116 | 001122 | DT3:  | .WORD  | TSTNUM, \$ERRPC, \$BDADR, \$GDDAT, \$BDDAT, 0 |         |                               |
| 10068 | 070314 | 001124 | 001126 | 000000 |       |        |                                               |         |                               |

|       |        |        |        |        |       |       |                                                              |
|-------|--------|--------|--------|--------|-------|-------|--------------------------------------------------------------|
| 10069 | 070322 | 001242 | 001116 | 001234 | DT5:  | .WORD | TSTNUM, \$ERRPC, PTNBR, \$BDADR, \$GDDAT, \$BDDAT, 0         |
| 10070 | 070330 | 001122 | 001124 | 001126 |       |       |                                                              |
| 10071 | 070336 | 000000 |        |        |       |       |                                                              |
| 10072 | 070340 | 001242 | 001116 | 001240 | DT6:  | .WORD | TSTNUM, \$ERRPC, OPPRT, \$BDADR, \$BDDAT, 0                  |
| 10073 | 070346 | 001122 | 001126 | 000000 |       |       |                                                              |
| 10074 | 070354 | 001242 | 001116 | 001236 | DT7:  | .WORD | TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$GDDAT, \$BDDAT, 0 |
| 10075 | 070362 | 001234 | 001122 | 001124 |       |       |                                                              |
| 10076 | 070370 | 001126 | 000000 |        |       |       |                                                              |
| 10077 | 070374 | 001242 | 001116 | 001236 | DT13: | .WORD | TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$BDDAT, 0          |
| 10078 | 070402 | 001234 | 001122 | 001126 |       |       |                                                              |
| 10079 | 070410 | 000000 |        |        |       |       |                                                              |
| 10080 | 070412 | 001242 | 001116 | 001236 | DT22: | .WORD | TSTNUM, \$ERRPC, SEIZPT, PTNBR, 0                            |
| 10081 | 070420 | 001234 | 000000 |        |       |       |                                                              |
| 10082 | 070424 | 001242 | 001116 | 001236 | DT23: | .WORD | TSTNUM, \$ERRPC, SEIZPT, \$BDADR, \$BDDAT, 0                 |
| 10083 | 070432 | 001122 | 001126 | 000000 |       |       |                                                              |
| 10084 | 070440 | 001242 | 001116 | 001236 | DT31: | .WORD | TSTNUM, \$ERRPC, SEIZPT, OPPRT, 0                            |
| 10085 | 070446 | 001240 | 000000 |        |       |       |                                                              |
| 10086 | 070452 | 001242 | 001116 | 001236 | DT36: | .WORD | TSTNUM, \$ERRPC, SEIZPT, 0                                   |
| 10087 | 070460 | 000000 |        |        |       |       |                                                              |
| 10088 | 070462 | 001242 | 001116 | 001234 | DT37: | .WORD | TSTNUM, \$ERRPC, PTNBR, 0                                    |
| 10089 | 070470 | 000000 |        |        |       |       |                                                              |
| 10090 | 070472 | 001242 | 001116 | 000000 | DT42: | .WORD | TSTNUM, \$ERRPC, 0                                           |
| 10091 | 070500 | 001242 | 001116 | 001170 | DT46: | .WORD | TSTNUM, \$ERRPC, \$TMP2, \$TMP3, 0                           |
| 10092 | 070506 | 001172 | 000000 |        |       |       |                                                              |
| 10093 | 070512 | 001242 | 001116 | 001240 | DT54: | .WORD | TSTNUM, \$ERRPC, OPPRT, SEIZPT, 0                            |
| 10094 | 070520 | 001236 | 000000 |        |       |       |                                                              |
| 10095 | 070524 | 001242 | 001116 | 001236 | DT55: | .WORD | TSTNUM, \$ERRPC, SEIZPT, TIME, 0                             |
| 10096 | 070532 | 001252 | 000000 |        |       |       |                                                              |
| 10097 | 070536 | 001300 | 000000 |        | DT56: | .WORD | \$RPADR, 0                                                   |
| 10098 |        |        |        |        |       |       |                                                              |
| 10099 | 070542 | 000    | 000    | 000    | DF1:  | .BYTE | 0,0,0,0,0                                                    |
| 10100 | 070545 | 000    | 000    |        |       |       |                                                              |
| 10101 | 070547 | 000    | 000    | 000    | DF5:  | .BYTE | 0,0,0,0,0,0                                                  |
| 10102 | 070552 | 000    | 000    | 000    |       |       |                                                              |
| 10103 | 070555 | 000    | 000    | 000    | DF7:  | .BYTE | 0,0,0,0,0,0,0                                                |
| 10104 | 070560 | 000    | 000    | 000    |       |       |                                                              |
| 10105 | 070563 | 000    |        |        |       |       |                                                              |
| 10106 | 070564 | 000    | 000    | 000    | DF31: | .BYTE | 0,0,0,0                                                      |
| 10107 | 070567 | 000    |        |        |       |       |                                                              |
| 10108 | 070570 | 000    | 000    | 000    | DF36: | .BYTE | 0,0,0                                                        |
| 10109 | 070573 | 000    | 000    |        | DF42: | .BYTE | 0,0                                                          |
| 10110 | 070575 | 000    | 000    | 000    | DF55: | .BYTE | 0,0,0,1                                                      |
| 10111 | 070600 | 001    |        |        |       |       |                                                              |
| 10112 | 070601 | 000    |        |        | DF56: | .BYTE | 0                                                            |
| 10113 |        |        |        |        |       |       |                                                              |
| 10114 |        |        |        |        |       | .EVEN |                                                              |
| 10115 |        |        |        |        |       |       |                                                              |
| 10116 |        |        |        |        |       |       |                                                              |
| 10117 |        |        |        |        |       |       |                                                              |
| 10118 |        |        |        |        |       |       |                                                              |
| 10119 |        |        |        |        |       |       |                                                              |
| 10120 |        |        |        |        |       |       |                                                              |
| 10121 |        |        |        |        |       |       |                                                              |
| 10122 |        |        |        |        |       |       |                                                              |
| 10123 |        |        |        |        |       |       |                                                              |
| 10124 |        |        |        |        |       |       |                                                              |

```

;:*****
.SBTTL CONSTANTS, TABLES, ETC
;:*****
:TABLE OF TEST STARTING ADDRESSES

```

|       |        |        |               |       |                             |    |
|-------|--------|--------|---------------|-------|-----------------------------|----|
| 10125 | 070602 | 003070 | TSTADR: .WORD | TST1  | :: STARTING ADDRESS OF TEST | 1  |
| 10126 | 070604 | 004470 | .WORD         | TST2  | :: STARTING ADDRESS OF TEST | 2  |
| 10127 | 070606 | 006070 | .WORD         | TST3  | :: STARTING ADDRESS OF TEST | 3  |
| 10128 | 070610 | 007470 | .WORD         | TST4  | :: STARTING ADDRESS OF TEST | 4  |
| 10129 | 070612 | 010626 | .WORD         | TST5  | :: STARTING ADDRESS OF TEST | 5  |
| 10130 | 070614 | 011764 | .WORD         | TST6  | :: STARTING ADDRESS OF TEST | 6  |
| 10131 | 070616 | 012754 | .WORD         | TST7  | :: STARTING ADDRESS OF TEST | 7  |
| 10132 | 070620 | 013744 | .WORD         | TST10 | :: STARTING ADDRESS OF TEST | 10 |
| 10133 | 070622 | 014366 | .WORD         | TST11 | :: STARTING ADDRESS OF TEST | 11 |
| 10134 | 070624 | 015010 | .WORD         | TST12 | :: STARTING ADDRESS OF TEST | 12 |
| 10135 | 070626 | 016366 | .WORD         | TST13 | :: STARTING ADDRESS OF TEST | 13 |
| 10136 | 070630 | 017744 | .WORD         | TST14 | :: STARTING ADDRESS OF TEST | 14 |
| 10137 | 070632 | 021246 | .WORD         | TST15 | :: STARTING ADDRESS OF TEST | 15 |
| 10138 | 070634 | 022550 | .WORD         | TST16 | :: STARTING ADDRESS OF TEST | 16 |
| 10139 | 070636 | 024010 | .WORD         | TST17 | :: STARTING ADDRESS OF TEST | 17 |
| 10140 | 070640 | 024676 | .WORD         | TST20 | :: STARTING ADDRESS OF TEST | 20 |
| 10141 | 070642 | 025564 | .WORD         | TST21 | :: STARTING ADDRESS OF TEST | 21 |
| 10142 | 070644 | 026744 | .WORD         | TST22 | :: STARTING ADDRESS OF TEST | 22 |
| 10143 | 070646 | 030124 | .WORD         | TST23 | :: STARTING ADDRESS OF TEST | 23 |
| 10144 | 070650 | 031134 | .WORD         | TST24 | :: STARTING ADDRESS OF TEST | 24 |
| 10145 | 070652 | 032144 | .WORD         | TST25 | :: STARTING ADDRESS OF TEST | 25 |
| 10146 | 070654 | 033352 | .WORD         | TST26 | :: STARTING ADDRESS OF TEST | 26 |
| 10147 | 070656 | 034560 | .WORD         | TST27 | :: STARTING ADDRESS OF TEST | 27 |
| 10148 | 070660 | 036554 | .WORD         | TST30 | :: STARTING ADDRESS OF TEST | 30 |
| 10149 | 070662 | 037250 | .WORD         | TST31 | :: STARTING ADDRESS OF TEST | 31 |
| 10150 | 070664 | 040510 | .WORD         | TST32 | :: STARTING ADDRESS OF TEST | 32 |
| 10151 | 070666 | 041750 | .WORD         | TST33 | :: STARTING ADDRESS OF TEST | 33 |
| 10152 | 070670 | 043040 | .WORD         | TST34 | :: STARTING ADDRESS OF TEST | 34 |
| 10153 | 070672 | 044130 | .WORD         | TST35 | :: STARTING ADDRESS OF TEST | 35 |
| 10154 | 070674 | 044744 | .WORD         | TST36 | :: STARTING ADDRESS OF TEST | 36 |
| 10155 | 070676 | 045560 | .WORD         | TST37 | :: STARTING ADDRESS OF TEST | 37 |
| 10156 | 070700 | 046542 | .WORD         | TST40 | :: STARTING ADDRESS OF TEST | 40 |
| 10157 | 070702 | 047524 | .WORD         | TST41 | :: STARTING ADDRESS OF TEST | 41 |
| 10158 | 070704 | 051132 | .WORD         | TST42 | :: STARTING ADDRESS OF TEST | 42 |
| 10159 | 070706 | 052540 | .WORD         | TST43 | :: STARTING ADDRESS OF TEST | 43 |
| 10160 | 070710 | 053422 | .WORD         | TST44 | :: STARTING ADDRESS OF TEST | 44 |
| 10161 | 070712 | 054310 | .WORD         | TST45 | :: STARTING ADDRESS OF TEST | 45 |
| 10162 | 070714 | 055256 | .WORD         | TST46 | :: STARTING ADDRESS OF TEST | 46 |
| 10163 |        |        |               |       |                             |    |
| 10164 |        |        |               |       |                             |    |
| 10165 |        |        |               |       |                             |    |
| 10166 | 070716 | 001    |               |       |                             |    |
| 10167 | 070717 | 002    |               |       |                             |    |
| 10168 | 070720 | 004    |               |       |                             |    |
| 10169 | 070721 | 010    |               |       |                             |    |
| 10170 | 070722 | 020    |               |       |                             |    |
| 10171 | 070723 | 040    |               |       |                             |    |
| 10172 | 070724 | 100    |               |       |                             |    |
| 10173 | 070725 | 200    |               |       |                             |    |
| 10174 |        |        |               |       |                             |    |
| 10175 | 070726 | 000046 | MAXTN: .WORD  | STN-1 | :: MAXIMUM TEST NUMBER      |    |
| 10176 |        |        |               |       |                             |    |
| 10177 |        | 000001 | .END          |       |                             |    |

:ATTENTION BIT TABLE

|               |     |                            |   |
|---------------|-----|----------------------------|---|
| ATABIT: .BYTE | 1   | :: ATTENTION BIT FOR DRIVE | 0 |
| .BYTE         | 2   | :: ATTENTION BIT FOR DRIVE | 1 |
| .BYTE         | 4   | :: ATTENTION BIT FOR DRIVE | 2 |
| .BYTE         | 10  | :: ATTENTION BIT FOR DRIVE | 3 |
| .BYTE         | 20  | :: ATTENTION BIT FOR DRIVE | 4 |
| .BYTE         | 40  | :: ATTENTION BIT FOR DRIVE | 5 |
| .BYTE         | 100 | :: ATTENTION BIT FOR DRIVE | 6 |
| .BYTE         | 200 | :: ATTENTION BIT FOR DRIVE | 7 |













|                 |       |       |       |       |       |       |       |       |       |       |       |       |       |  |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| MSTCK = 000010  | 1303# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| MWR = 000040    | 1305# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| MXF = 001000    | 1232# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NBA = 100000    | 1336# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NED = 010000    | 1235# | 2135  | 2139  | 2156  | 2160  |       |       |       |       |       |       |       |       |  |
| NEM = 004000    | 1234# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NHS = 002000    | 1369# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NOATA = 000001  | 1931# | 1387# |       |       |       |       |       |       |       |       |       |       |       |  |
|                 | 2444  | 2450  | 2681  | 2687  | 2934  | 2940  | 3113  | 3119  | 3208  | 3214  | 3353  | 3359  |       |  |
|                 | 3485  | 3491  | 3565  | 3571  | 3770  | 3776  | 3974  | 3980  | 4156  | 4162  | 4357  | 4363  | 4391# |  |
|                 | 4516  | 4521  | 4684  | 4689  | 4806  | 4811  | 4876# | 4943# | 4987  | 4992  | 5057# | 5124# | 5168  |  |
|                 | 5173  | 5318  | 5323  | 5467  | 5472  | 5657  | 5662  | 5846  | 5851  | 6022  | 6027  | 6128  | 6133  |  |
|                 | 6223  | 6228  | 6359# | 6396  | 6402  | 6407# | 6561# | 6598  | 6604  | 6609# | 6798  | 6803  | 6967  |  |
|                 | 6972  | 7096  | 7101  | 7233  | 7238  | 7393  | 7398  | 7541  | 7546  | 7699  | 7704  | 7913  | 7918  |  |
|                 | 8039# | 8083  | 8089  | 8094# | 8153# | 8197  | 8203  | 8208# | 8371  | 8376  | 8536  | 8541  |       |  |
|                 | 9449# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NOCLC 062235    | 2028  |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NOSEIZ 001246   | 1566# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| NTRH11 062444   | 2073  | 9473# |       |       |       |       |       |       |       |       |       |       |       |  |
| OCYL = 100000   | 1422# | 1432# |       |       |       |       |       |       |       |       |       |       |       |  |
| OFREV = 000200  | 1399# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF100 = 000004  | 1395# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF200 = 000010  | 1396# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF25 = 000001   | 1393# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF400 = 000020  | 1397# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF50 = 000002   | 1394# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OF800 = 000040  | 1398# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OPE = 020000    | 1430# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OPI = 020000    | 1294# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| OPPRT 001240    | 1563# | 2340* | 2577* | 3164* | 3309* | 3619* | 3823* | 4085* | 4286* | 4617* | 4739* | 4867* | 5048* |  |
|                 | 5228# | 5377* | 5530* | 5719* | 5949* | 6055* | 6278* | 6480* | 6689* | 6858* | 7025* | 7162* | 7606* |  |
|                 | 7820* | 8269* | 8434* | 10072 | 10084 | 10093 |       |       |       |       |       |       |       |  |
| OR = 000200     | 1230# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PAR = 000010    | 1284# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PAT = 000020    | 1227# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PGE = 002000    | 1233# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PGM = 001000    | 1271# | 2216  | 2218  | 2222  | 2248  | 2250  | 2254  | 2353  | 2415  | 2590  | 2652  | 2831  | 2871  |  |
|                 | 2905  | 3010  | 3050  | 3084  | 3179  | 3324  | 3456  | 3536  | 3632  | 3673  | 3716  | 3739  | 3836  |  |
|                 | 3877  | 3920  | 3943  | 4125  | 4326  | 4630  | 4653  | 4752  | 4775  | 4889  | 4956  | 5070  | 5137  |  |
|                 | 5241  | 5287  | 5390  | 5436  | 5543  | 5595  | 5626  | 5732  | 5784  | 5815  | 5937  | 5960  | 5991  |  |
|                 | 6043  | 6066  | 6097  | 6192  | 6767  | 6936  | 7065  | 7202  | 7362  | 7510  | 7637  | 7668  | 7851  |  |
|                 | 7882  | 8052  | 8166  | 8340  | 8505  |       |       |       |       |       |       |       |       |  |
| PIP = 020000    | 1275# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PIRQ = 177772   | 1102# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PIRGVE = 000240 | 1196# |       |       |       |       |       |       |       |       |       |       |       |       |  |
| PLU = 020000    | 1372# | 1389# |       |       |       |       |       |       |       |       |       |       |       |  |
| PORTA 001224    | 1557# | 1994  | 1998  | 2001  | 2005  | 2017  | 2020  | 2126  | 2127  | 2172  | 2173  | 2210  |       |  |
|                 | 2211  | 2335  | 2336  | 2350  | 2351  | 2416  | 2435  | 2437  | 2443  | 2575  | 2576  | 2577  |       |  |
|                 | 2603  | 2604  | 2653  | 2672  | 2674  | 2680  | 2697  | 2795  | 2796  | 2803  | 2825  | 2826  | 2906  |  |
|                 | 2925  | 2927  | 2933  | 2992  | 2993  | 3061  | 3062  | 3085  | 3104  | 3106  | 3112  | 3161  | 3162  |  |
|                 | 3170  | 3171  | 3180  | 3199  | 3201  | 3207  | 3222  | 3223  | 3309  | 3325  | 3344  | 3346  | 3352  |  |
|                 | 3384  | 3385  | 3408  | 3409  | 3438  | 3439  | 3440  | 3457  | 3476  | 3478  | 3484  | 3537  | 3556  |  |
|                 | 3558  | 3564  | 3614  | 3615  | 3629  | 3630  | 3667  | 3668  | 3710  | 3711  | 3730  | 3731  | 3740  |  |
|                 | 3759  | 3761  | 3769  | 3821  | 3822  | 3823  | 3854  | 3855  | 3897  | 3898  | 3944  | 3963  | 3965  |  |
|                 | 3973  | 4022  | 4030  | 4037  | 4038  | 4082  | 4083  | 4091  | 4116  | 4117  | 4126  | 4145  | 4147  |  |
|                 | 4155  | 4223  | 4231  | 4258  | 4259  | 4286  | 4327  | 4346  | 4348  | 4356  | 4372  | 4373  | 4422  |  |
|                 | 4430  | 4437  | 4438  | 4486  | 4505  | 4507  | 4515  | 4536  | 4537  | 4599  | 4612  | 4613  | 4627  |  |
|                 | 4628  | 4644  | 4645  | 4654  | 4673  | 4675  | 4683  | 4721  | 4737  | 4738  | 4739  | 4776  | 4795  |  |

|       |       |       |       |       |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|
| 4797  | 4805  | 4851  | 4867  | 4868  | 4869 | 4892 | 4893 | 4903 | 4925 | 4926 | 4947 | 4948 |
| 4957  | 4976  | 4978  | 4986  | 5032  | 5045 | 5046 | 5063 | 5064 | 5076 | 5077 | 5089 | 5090 |
| 5138  | 5157  | 5159  | 5167  | 5210  | 5226 | 5227 | 5228 | 5250 | 5251 | 5288 | 5307 | 5309 |
| 5317  | 5359  | 5372  | 5373  | 5387  | 5388 | 5427 | 5428 | 5437 | 5456 | 5458 | 5466 | 5512 |
| 5528  | 5529  | 5530  | 5556  | 5557  | 5598 | 5599 | 5609 | 5617 | 5618 | 5627 | 5646 | 5648 |
| 5656  | 5701  | 5714  | 5715  | 5729  | 5730 | 5777 | 5778 | 5790 | 5791 | 5816 | 5835 | 5837 |
| 5845  | 5887  | 5900  | 5906  | 5907  | 5949 | 5963 | 5964 | 5974 | 5982 | 5983 | 5992 | 6011 |
| 6013  | 6021  | 6037  | 6038  | 6054  | 6059 | 6060 | 6072 | 6073 | 6098 | 6117 | 6119 | 6127 |
| 6166  | 6193  | 6212  | 6214  | 6222  | 6263 | 6275 | 6276 | 6302 | 6303 | 6366 | 6385 | 6387 |
| 6395  | 6431  | 6465  | 6480  | 6492  | 6493 | 6568 | 6587 | 6589 | 6597 | 6613 | 6614 | 6674 |
| 6686  | 6687  | 6715  | 6716  | 6736  | 6737 | 6758 | 6759 | 6768 | 6787 | 6789 | 6797 | 6843 |
| 6858  | 6893  | 6894  | 6937  | 6956  | 6958 | 6966 | 7009 | 7022 | 7023 | 7066 | 7085 | 7087 |
| 7095  | 7146  | 7162  | 7186  | 7187  | 7203 | 7222 | 7224 | 7232 | 7286 | 7294 | 7295 | 7296 |
| 7353  | 7354  | 7363  | 7382  | 7384  | 7392 | 7434 | 7475 | 7476 | 7511 | 7530 | 7532 | 7540 |
| 7592  | 7603  | 7604  | 7613  | 7614  | 7624 | 7625 | 7630 | 7631 | 7643 | 7644 | 7669 | 7688 |
| 7690  | 7698  | 7806  | 7820  | 7824  | 7825 | 7833 | 7834 | 7854 | 7855 | 7865 | 7873 | 7874 |
| 7883  | 7902  | 7904  | 7912  | 8009  | 8017 | 8053 | 8072 | 8074 | 8082 | 8123 | 8134 | 8143 |
| 8144  | 8145  | 8157  | 8158  | 8167  | 8186 | 8188 | 8196 | 8253 | 8269 | 8270 | 8271 | 8312 |
| 8331  | 8332  | 8341  | 8360  | 8362  | 8370 | 8418 | 8431 | 8432 | 8442 | 8443 | 8506 | 8525 |
| 8527  | 8535  |       |       |       |      |      |      |      |      |      |      |      |
| 2004  | 9441# |       |       |       |      |      |      |      |      |      |      |      |
| 1558# | 1998# | 1999* | 2000* | 2003* | 2011 | 2147 | 2148 | 2189 | 2190 | 2242 | 2243 | 2280 |
| 2338  | 2339  | 2340  | 2366  | 2367  | 2420 | 2431 | 2432 | 2449 | 2460 | 2572 | 2573 | 2587 |
| 2588  | 2657  | 2668  | 2669  | 2686  | 2813 | 2814 | 2882 | 2883 | 2910 | 2921 | 2922 | 2939 |
| 2974  | 2975  | 2982  | 3004  | 3005  | 3089 | 3100 | 3101 | 3118 | 3164 | 3184 | 3195 | 3196 |
| 3213  | 3239  | 3240  | 3263  | 3264  | 3306 | 3307 | 3315 | 3316 | 3329 | 3340 | 3341 | 3358 |
| 3367  | 3368  | 3461  | 3472  | 3473  | 3490 | 3518 | 3519 | 3520 | 3541 | 3552 | 3553 | 3570 |
| 3617  | 3618  | 3619  | 3650  | 3651  | 3693 | 3694 | 3744 | 3755 | 3756 | 3775 | 3818 | 3819 |
| 3833  | 3834  | 3871  | 3872  | 3914  | 3915 | 3934 | 3935 | 3948 | 3959 | 3960 | 3979 | 4025 |
| 4057  | 4058  | 4085  | 4130  | 4141  | 4142 | 4161 | 4171 | 4172 | 4226 | 4238 | 4239 | 4283 |
| 4284  | 4292  | 4317  | 4318  | 4331  | 4342 | 4343 | 4362 | 4425 | 4457 | 4458 | 4490 | 4501 |
| 4502  | 4520  | 4553  | 4554  | 4603  | 4615 | 4616 | 4617 | 4658 | 4669 | 4670 | 4688 | 4725 |
| 4734  | 4735  | 4749  | 4750  | 4766  | 4767 | 4780 | 4791 | 4792 | 4810 | 4855 | 4864 | 4865 |
| 4882  | 4883  | 4895  | 4896  | 4908  | 4909 | 4961 | 4972 | 4973 | 4991 | 5036 | 5048 | 5049 |
| 5050  | 5073  | 5074  | 5084  | 5106  | 5107 | 5128 | 5129 | 5142 | 5153 | 5154 | 5172 | 5214 |
| 5223  | 5224  | 5238  | 5239  | 5278  | 5279 | 5292 | 5303 | 5304 | 5322 | 5363 | 5375 | 5376 |
| 5377  | 5399  | 5400  | 5441  | 5452  | 5453 | 5471 | 5516 | 5525 | 5526 | 5540 | 5541 | 5588 |
| 5589  | 5601  | 5602  | 5631  | 5642  | 5643 | 5661 | 5705 | 5717 | 5718 | 5719 | 5745 | 5746 |
| 5787  | 5788  | 5798  | 5806  | 5807  | 5820 | 5831 | 5832 | 5850 | 5891 | 5914 | 5915 | 5921 |
| 5922  | 5948  | 5953  | 5954  | 5966  | 5967 | 5996 | 6007 | 6008 | 6026 | 6055 | 6069 | 6070 |
| 6080  | 6088  | 6089  | 6102  | 6113  | 6114 | 6132 | 6170 | 6197 | 6208 | 6209 | 6227 | 6267 |
| 6278  | 6290  | 6291  | 6370  | 6381  | 6382 | 6401 | 6411 | 6412 | 6469 | 6477 | 6478 | 6504 |
| 6505  | 6572  | 6583  | 6584  | 6603  | 6633 | 6678 | 6689 | 6724 | 6725 | 6772 | 6783 | 6784 |
| 6802  | 6847  | 6855  | 6856  | 6884  | 6885 | 6905 | 6906 | 6927 | 6928 | 6941 | 6952 | 6953 |
| 6971  | 7013  | 7025  | 7049  | 7050  | 7070 | 7081 | 7082 | 7100 | 7150 | 7159 | 7160 | 7207 |
| 7218  | 7219  | 7237  | 7290  | 7327  | 7328 | 7367 | 7378 | 7379 | 7397 | 7438 | 7442 | 7443 |
| 7444  | 7501  | 7502  | 7515  | 7526  | 7527 | 7545 | 7596 | 7606 | 7610 | 7611 | 7619 | 7620 |
| 7640  | 7641  | 7651  | 7659  | 7660  | 7673 | 7684 | 7685 | 7703 | 7810 | 7817 | 7818 | 7827 |
| 7828  | 7838  | 7839  | 7844  | 7845  | 7857 | 7858 | 7887 | 7898 | 7899 | 7917 | 8013 | 8020 |
| 8029  | 8030  | 8031  | 8043  | 8044  | 8057 | 8068 | 8069 | 8088 | 8127 | 8131 | 8171 | 8182 |
| 8183  | 8202  | 8257  | 8266  | 8267  | 8277 | 8278 | 8345 | 8356 | 8357 | 8375 | 8422 | 8434 |
| 8435  | 8436  | 8477  | 8496  | 8497  | 8510 | 8521 | 8522 | 8540 |      |      |      |      |
| 2010  | 9445# |       |       |       |      |      |      |      |      |      |      |      |
| 1559# | 2017* | 2018* | 2019* | 5899  | 6174 |      |      |      |      |      |      |      |
| 1418# |       |       |       |       |      |      |      |      |      |      |      |      |
| 1119# |       |       |       |       |      |      |      |      |      |      |      |      |

PORTAI 062161  
PORTB 001226

PORTBI 062207  
PORTC 001230  
PRE = 000020  
PRO = 000000

PR1 = 000040  
PR2 = 000100  
PR3 = 000140  
PR4 = 000200  
PR5 = 000240  
PR6 = 000300  
PR7 = 000340  
PS = 177776  
PSEL = 002000  
PSU = 000001  
PSW = 177776  
PTNBR = 001234

|       |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|--|
| 1120# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1121# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1122# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1123# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1124# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1125# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1126# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1099# | 1100  | 2035* | 2320* | 2391* | 2408* | 2557* | 2628* | 2645* |       |       |       |       |  |  |
| 1210# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1415# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1100# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1561# | 2127* | 2148* | 2173* | 2190* | 2211* | 2243* | 2339* | 2351* | 2367* | 2431* | 2435* | 2443* |  |  |
| 2449# | 2460* | 2576* | 2588* | 2604* | 2668* | 2672* | 2680* | 2686* | 2697* | 2796* | 2814* | 2826* |  |  |
| 2883# | 2921* | 2925* | 2933* | 2939* | 2975* | 2993* | 3005* | 3062* | 3100* | 3104* | 3112* | 3118* |  |  |
| 3171# | 3195* | 3199* | 3207* | 3213* | 3223* | 3240* | 3264* | 3316* | 3340* | 3344* | 3352* | 3358* |  |  |
| 3368# | 3385* | 3409* | 3439* | 3472* | 3476* | 3484* | 3490* | 3519* | 3552* | 3556* | 3564* | 3570* |  |  |
| 3618# | 3630* | 3651* | 3668* | 3694* | 3711* | 3731* | 3755* | 3759* | 3769* | 3775* | 3822* | 3834* |  |  |
| 3855# | 3872* | 3898* | 3915* | 3935* | 3959* | 3963* | 3973* | 3979* | 4038* | 4058* | 4117* | 4141* |  |  |
| 4145# | 4155* | 4161* | 4172* | 4239* | 4259* | 4318* | 4342* | 4346* | 4356* | 4362* | 4373* | 4438* |  |  |
| 4458# | 4501* | 4505* | 4515* | 4520* | 4537* | 4554* | 4616* | 4628* | 4645* | 4669* | 4673* | 4683* |  |  |
| 4688# | 4738* | 4750* | 4767* | 4791* | 4795* | 4805* | 4810* | 4869* | 4883* | 4893* | 4896* | 4903* |  |  |
| 4909# | 4926* | 4948* | 4972* | 4976* | 4986* | 4991* | 5050* | 5064* | 5074* | 5077* | 5084* | 5090* |  |  |
| 5107# | 5129* | 5153* | 5157* | 5167* | 5172* | 5227* | 5239* | 5251* | 5279* | 5303* | 5307* | 5317* |  |  |
| 5322# | 5376* | 5388* | 5400* | 5428* | 5452* | 5456* | 5466* | 5471* | 5529* | 5541* | 5557* | 5589* |  |  |
| 5599# | 5602* | 5609* | 5618* | 5642* | 5646* | 5656* | 5661* | 5718* | 5730* | 5746* | 5778* | 5788* |  |  |
| 5791# | 5798* | 5807* | 5831* | 5835* | 5845* | 5850* | 5907* | 5915* | 5922* | 5954* | 5964* | 5967* |  |  |
| 5974# | 5983* | 6007* | 6011* | 6021* | 6026* | 6038* | 6060* | 6070* | 6073* | 6080* | 6089* | 6113* |  |  |
| 6117# | 6127* | 6132* | 6208* | 6212* | 6222* | 6227* | 6291* | 6303* | 6381* | 6385* | 6395* | 6401* |  |  |
| 6412# | 6493* | 6505* | 6583* | 6587* | 6597* | 6603* | 6614* | 6716* | 6725* | 6737* | 6759* | 6783* |  |  |
| 6787# | 6797* | 6802* | 6885* | 6894* | 6906* | 6928* | 6952* | 6956* | 6971* | 7050* | 7081* | 7081* |  |  |
| 7085# | 7095* | 7100* | 7187* | 7218* | 7222* | 7232* | 7237* | 7295* | 7328* | 7354* | 7378* | 7382* |  |  |
| 7392# | 7397* | 7443* | 7476* | 7502* | 7526* | 7530* | 7540* | 7545* | 7611* | 7614* | 7620* | 7625* |  |  |
| 7631# | 7641* | 7644* | 7651* | 7660* | 7684* | 7688* | 7698* | 7703* | 7825* | 7828* | 7834* | 7839* |  |  |
| 7845# | 7855* | 7858* | 7865* | 7874* | 7898* | 7902* | 7912* | 7917* | 8030* | 8044* | 8068* | 8072* |  |  |
| 8082# | 8088* | 8144* | 8158* | 8182* | 8186* | 8196* | 8202* | 8271* | 8278* | 8312* | 8332* | 8356* |  |  |
| 8360# | 8370* | 8375* | 8436* | 8443* | 8477* | 8497* | 8521* | 8525* | 8535* | 8540* | 10065 | 10069 |  |  |
| 10074 | 10077 | 10080 | 10088 |       |       |       |       |       |       |       |       |       |  |  |
| 1191# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1381# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 9216  | 9408# |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 9289  | 9409# |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1992  | 2048  | 2074  | 9410# |       |       |       |       |       |       |       |       |       |  |  |
| 1207# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1567# | 2412* | 2440* | 2649* | 2677* | 2902* | 2930* | 3081* | 3109* | 3176* | 3204* | 3219  | 3321* |  |  |
| 3349# | 3364  | 3453* | 3481* | 3533* | 3561* | 3736* | 3764* | 3940* | 3968* | 4122* | 4150* | 4323* |  |  |
| 4351# | 4482* | 4510* | 4525  | 4650* | 4678* | 4772* | 4800* | 4888* | 4953* | 4981* | 5069* | 5134* |  |  |
| 5162# | 5284* | 5312* | 5433* | 5461* | 5594* | 5623* | 5651* | 5783* | 5812* | 5840* | 5959* | 5988* |  |  |
| 6016# | 6065* | 6094* | 6122* | 6189* | 6217* | 6362* | 6390* | 6564* | 6592* | 6764* | 6792* | 6930* |  |  |
| 6961# | 7062* | 7090* | 7199* | 7227* | 7359* | 7387* | 7507* | 7535* | 7636* | 7665* | 7693* | 7950* |  |  |
| 7879# | 7907* | 8049* | 8077* | 8163* | 8191* | 8337* | 8365* | 8502* | 8530* |       |       |       |  |  |
| 1931# | 2441  | 2678  | 2931  | 3110  | 3205  | 3350  | 3482  | 3562  | 3578* | 3765  | 3969  | 4151  |  |  |
| 4352  | 4511  | 4679  | 4801  | 4982  | 5163  | 5313  | 5462  | 5652  | 5841  | 6017  | 6123  | 6218  |  |  |
| 6391  | 6593  | 6793  | 6962  | 7091  | 7228  | 7388  | 7536  | 7694  | 7908  | 8078  | 8192  | 8366  |  |  |
| 8531  |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 9412# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1186# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |

PWRVEC = 000024  
RAW = 000020  
RDCHR = 104410  
RDLIN = 104411  
RDOCT = 104412  
RDY = 000200  
RELEA = 001250

RELOK = 000001

RESREG = 104414  
RESVEC = 000010

# M15

CZRJEBO, DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 195  
 CZRJEB.P11 04-NOV-77 13:27

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0194

RMR = 000004  
 RPAS = 000016  
 RPBA = 000004  
 RPCA = 000034  
 RPCC = 000036  
 RPCS1 = 000000

|       |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|--|
| 1283* |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1454* | 5905* | 6181* | 7618* | 7832* | 8023  | 8033  | 8137  | 8147  |       |       |       |       |  |  |
| 1449* |       |       |       |       |       |       |       |       |       |       |       |       |  |  |
| 1461* | 2373  | 2497  | 2610  | 2734  |       |       |       |       |       |       |       |       |  |  |
| 1462* | 2372  | 2502  | 2609  | 2739  |       |       |       |       |       |       |       |       |  |  |
| 1447* | 2808* | 2843  | 2844  | 2861* | 2987* | 3022  | 3023  | 3040* | 3172* | 3317* | 3445* | 3525* |  |  |
| 3645* | 3732* | 3765* | 3766* | 3849* | 3936* | 3969* | 3970* | 4118* | 4151* | 4152* | 4319* | 4352* |  |  |
| 4353* | 4511* | 4512* | 4601* | 4602* | 4605* | 4606* | 4614  | 4646* | 4679* | 4680* | 4723* | 4724* |  |  |
| 4727* | 4728* | 4736  | 4768* | 4801* | 4802* | 4853* | 4854* | 4857* | 4858* | 4884* | 4949* | 4982* |  |  |
| 4983* | 5034* | 5035* | 5038* | 5039* | 5065* | 5130* | 5163* | 5164* | 5212* | 5213* | 5216* | 5217* |  |  |
| 5225  | 5257  | 5258  | 5280* | 5313* | 5314* | 5361* | 5362* | 5365* | 5366* | 5374  | 5406  | 5407  |  |  |
| 5429* | 5462* | 5463* | 5514* | 5515* | 5518* | 5519* | 5558* | 5572  | 5573  | 5578* | 5590* | 5619* |  |  |
| 5652* | 5653* | 5703* | 5704* | 5707* | 5708* | 5747* | 5761  | 5762  | 5767* | 5779* | 5808* | 5841* |  |  |
| 5842* | 5889* | 5890* | 5893* | 5894* | 5955* | 5984* | 6017* | 6018* | 6051* | 6090* | 6123* | 6124* |  |  |
| 6168* | 6169* | 6172* | 6173* | 6218* | 6219* | 6265* | 6266* | 6269* | 6270* | 6391* | 6392* | 6433* |  |  |
| 6434* | 6467* | 6468* | 6471* | 6472* | 6593* | 6594* | 6635* | 6636* | 6676* | 6677* | 6680* | 6681* |  |  |
| 6695* | 6697  | 6698  | 6717* | 6752* | 6760* | 6793* | 6794* | 6845* | 6846* | 6849* | 6850* | 6964* |  |  |
| 6866  | 6867  | 6886* | 6921* | 6929* | 6962* | 6963* | 7011* | 7012* | 7015* | 7016* | 7091* | 7092* |  |  |
| 7148* | 7149* | 7152* | 7153* | 7228* | 7229* | 7288* | 7289* | 7292* | 7293* | 7301* | 7355* | 7363* |  |  |
| 7389* | 7436* | 7437* | 7440* | 7441* | 7449* | 7503* | 7536* | 7537* | 7594* | 7595* | 7598* | 7599* |  |  |
| 7632* | 7661* | 7694* | 7695* | 7723  | 7724  | 7729* | 7740  | 7741  | 7746* | 7757  | 7758  | 7763* |  |  |
| 7808* | 7809* | 7812* | 7813* | 7846* | 7875* | 7908* | 7909* | 7937  | 7938  | 7943* | 7954  | 7955  |  |  |
| 7960* | 7971  | 7972  | 7977* | 8011* | 8012* | 8015* | 8016* | 8045* | 8078* | 8079* | 8125* | 8126* |  |  |
| 8129* | 8130* | 8159* | 8192* | 8193* | 8255* | 8256* | 8259* | 8260* | 8283* | 8333* | 8366* | 8367* |  |  |
| 8420* | 8421* | 8424* | 8425* | 8448* | 8498* | 8531* | 8532* |       |       |       |       |       |  |  |
| 1451* | 2126* | 2130  | 2131  | 2146* | 2147* | 2151  | 2152  | 2167* | 2172* | 2189* | 2210* | 2242* |  |  |
| 2278* | 2280* | 2335* | 2338* | 2350* | 2366* | 2416* | 2420* | 2432* | 2437* | 2572* | 2575* | 2587* |  |  |
| 2603* | 2653* | 2657* | 2669* | 2674* | 2795* | 2813* | 2825* | 2882* | 2906* | 2910* | 2922* | 2927* |  |  |
| 2974* | 2992* | 3004* | 3061* | 3085* | 3089* | 3101* | 3106* | 3161* | 3170* | 3180* | 3184* | 3198* |  |  |
| 3201* | 3222* | 3239* | 3263* | 3306* | 3315* | 3329* | 3341* | 3346* | 3347* | 3384* | 3384* | 3408* |  |  |
| 3438* | 3457* | 3461* | 3473* | 3478* | 3518* | 3537* | 3541* | 3553* | 3558* | 3614* | 3617* | 3629* |  |  |
| 3650* | 3667* | 3688* | 3693* | 3710* | 3730* | 3740* | 3744* | 3756* | 3761* | 3818* | 3821* | 3833* |  |  |
| 3854* | 3871* | 3892* | 3897* | 3914* | 3934* | 3944* | 3948* | 3960* | 3965* | 4022* | 4025* | 4030* |  |  |
| 4037* | 4057* | 4082* | 4090* | 4091* | 4116* | 4126* | 4130* | 4142* | 4147* | 4171* | 4223* | 4226* |  |  |
| 4231* | 4238* | 4258* | 4283* | 4291* | 4292* | 4317* | 4327* | 4331* | 4343* | 4372* | 4372* | 4422* |  |  |
| 4425* | 4430* | 4437* | 4457* | 4486* | 4490* | 4502* | 4507* | 4531* | 4536* | 4553* | 4599* | 4603* |  |  |
| 4612* | 4615* | 4627* | 4644* | 4654* | 4658* | 4670* | 4675* | 4721* | 4725* | 4734* | 4737* | 4749* |  |  |
| 4766* | 4776* | 4780* | 4792* | 4797* | 4851* | 4855* | 4864* | 4868* | 4882* | 4892* | 4895* | 4908* |  |  |
| 4925* | 4947* | 4957* | 4961* | 4973* | 4978* | 5032* | 5036* | 5045* | 5049* | 5063* | 5073* | 5076* |  |  |
| 5089* | 5106* | 5128* | 5138* | 5142* | 5154* | 5159* | 5210* | 5214* | 5223* | 5226* | 5238* | 5250* |  |  |
| 5278* | 5288* | 5292* | 5304* | 5309* | 5359* | 5363* | 5372* | 5375* | 5387* | 5399* | 5427* | 5437* |  |  |
| 5441* | 5453* | 5458* | 5512* | 5516* | 5525* | 5528* | 5540* | 5556* | 5588* | 5598* | 5601* | 5617* |  |  |
| 5627* | 5631* | 5643* | 5648* | 5701* | 5705* | 5714* | 5717* | 5729* | 5745* | 5777* | 5787* | 5790* |  |  |
| 5806* | 5816* | 5820* | 5832* | 5837* | 5887* | 5891* | 5899* | 5906* | 5914* | 5921* | 5953* | 5963* |  |  |
| 5966* | 5982* | 5992* | 5996* | 6008* | 6013* | 6037* | 6059* | 6069* | 6072* | 6088* | 6098* | 6102* |  |  |
| 6114* | 6119* | 6166* | 6170* | 6174* | 6193* | 6197* | 6209* | 6214* | 6263* | 6267* | 6275* | 6290* |  |  |
| 6302* | 6366* | 6370* | 6382* | 6387* | 6411* | 6431* | 6465* | 6469* | 6477* | 6492* | 6504* | 6568* |  |  |
| 6572* | 6584* | 6589* | 6613* | 6633* | 6674* | 6678* | 6686* | 6714* | 6715* | 6724* | 6736* | 6758* |  |  |
| 6768* | 6772* | 6784* | 6789* | 6843* | 6847* | 6855* | 6883* | 6884* | 6893* | 6905* | 6927* | 6937* |  |  |
| 6941* | 6953* | 6958* | 7009* | 7013* | 7022* | 7049* | 7066* | 7070* | 7082* | 7087* | 7146* | 7150* |  |  |
| 7159* | 7186* | 7203* | 7207* | 7219* | 7224* | 7286* | 7290* | 7294* | 7327* | 7353* | 7363* | 7367* |  |  |
| 7379* | 7384* | 7434* | 7438* | 7442* | 7475* | 7501* | 7511* | 7515* | 7527* | 7532* | 7592* | 7596* |  |  |
| 7603* | 7610* | 7613* | 7619* | 7624* | 7630* | 7640* | 7643* | 7659* | 7659* | 7673* | 7685* | 7690* |  |  |
| 7806* | 7810* | 7817* | 7824* | 7827* | 7833* | 7838* | 7844* | 7854* | 7857* | 7873* | 7883* | 7887* |  |  |
| 7899* | 7904* | 8009* | 8013* | 8017* | 8020* | 8029* | 8043* | 8053* | 8057* | 8069* | 8074* | 8123* |  |  |
| 8127* | 8131* | 8134* | 8143* | 8157* | 8167* | 8171* | 8183* | 8188* | 8253* | 8257* | 8266* | 8270* |  |  |

RPCS2 = 000010

RPDA = 000006  
RPDB = 000022  
RPDS1 = 000012

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 8277* | 8331* | 8341* | 8345* | 8357* | 8362* | 8418* | 8422* | 8431* | 8435* | 8442* | 8496* | 8506* |
| 8510* | 8522* | 8527* |       | 2709  |       |       |       |       |       |       |       |       |
| 1450* | 2378  | 2472  | 2615  |       |       |       |       |       |       |       |       |       |
| 1456* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1452* | 2128  | 2149  | 2213  | 2214  | 2228  | 2229  | 2245  | 2246  | 2260  | 2261  | 2337* | 2341  |
| 2343* | 2352* | 2385  | 2413  | 2417  | 2421  | 2574* | 2578  | 2580  | 2589  | 2622  | 2650  | 2654  |
| 2658  | 2816  | 2817  | 2828  | 2829  | 2868  | 2869  | 2888  | 2903  | 2907  | 2911  | 2995  | 2996  |
| 3007  | 3008  | 3047  | 3048  | 3067  | 3082  | 3086  | 3090  | 3163* | 3177  | 3181  | 3185  | 3225  |
| 3226  | 3242  | 3243  | 3265  | 3308* | 3322  | 3326  | 3330  | 3370  | 3371  | 3387  | 3388  | 3410  |
| 3454  | 3458  | 3462  | 3534  | 3538* | 3542  | 3616* | 3620  | 3622  | 3631  | 3653  | 3654  | 3670  |
| 3671  | 3696  | 3697  | 3713  | 3714  | 3737  | 3741  | 3745  | 3820* | 3824  | 3826  | 3835  | 3857  |
| 3858  | 3874  | 3875  | 3900  | 3901  | 3917  | 3918  | 3941  | 3945  | 3949  | 4026  | 4031  | 4040  |
| 4041  | 4060  | 4061  | 4084* | 4097  | 4098  | 4123  | 4127  | 4131  | 4174  | 4175  | 4227  | 4232  |
| 4241  | 4242  | 4261  | 4262  | 4285* | 4298  | 4299  | 4324  | 4328  | 4332  | 4375  | 4376  | 4426  |
| 4431  | 4440  | 4441  | 4460  | 4461  | 4483  | 4487  | 4491  | 4539  | 4540  | 4556  | 4557  | 4600* |
| 4604* | 4618  | 4620  | 4629  | 4651  | 4655  | 4659  | 4722* | 4726* | 4740  | 4742  | 4751  | 4773  |
| 4777  | 4781  | 4852* | 4856* | 4866* | 4874* | 4890  | 4894  | 4897  | 4911  | 4912  | 4928  | 4929  |
| 4954  | 4958  | 4962* | 5033* | 5037* | 5047* | 5055* | 5071  | 5075  | 5078  | 5092  | 5093  | 5109  |
| 5110  | 5135  | 5139  | 5143  | 5211* | 5215* | 5229  | 5231  | 5240  | 5285  | 5289  | 5293  | 5360* |
| 5364* | 5378  | 5380  | 5389  | 5434  | 5438  | 5442  | 5513* | 5517* | 5527* | 5531  | 5533  | 5542  |
| 5564  | 5565  | 5596  | 5600  | 5603  | 5624  | 5628  | 5632* | 5702* | 5706* | 5716* | 5720  | 5722  |
| 5731  | 5753  | 5754  | 5785  | 5789  | 5792  | 5813  | 5817  | 5821  | 5888* | 5892* | 5912  | 5916  |
| 5923  | 5934  | 5935  | 5961  | 5965  | 5968  | 5989  | 5993  | 5997  | 6040  | 6041  | 6067  | 6071  |
| 6074  | 6095  | 6099  | 6103  | 6167* | 6171* | 6190  | 6194  | 6198  | 6264* | 6268* | 6277* | 6296  |
| 6309  | 6310  | 6342  | 6343  | 6363  | 6367  | 6371  | 6414  | 6415  | 6432* | 6466* | 6470* | 6479* |
| 6498  | 6511  | 6512  | 6544  | 6545  | 6565  | 6569  | 6573  | 6616  | 6617  | 6634* | 6675* | 6679* |
| 6688* | 6727  | 6728  | 6765  | 6769  | 6773  | 6844* | 6848* | 6857* | 6896  | 6897  | 6934  | 6938  |
| 6942  | 7010* | 7014* | 7024* | 7048  | 7051  | 7063  | 7067  | 7071  | 7147* | 7151* | 7161* | 7185  |
| 7188  | 7200  | 7204  | 7208  | 7287* | 7291* | 7306  | 7313  | 7314  | 7334  | 7335  | 7360  | 7364  |
| 7368  | 7435* | 7439* | 7454  | 7461  | 7462  | 7482  | 7483  | 7508  | 7512  | 7516  | 7593* | 7597* |
| 7605* | 7638  | 7642  | 7645  | 7666  | 7670  | 7674  | 7807* | 7811* | 7819* | 7852  | 7856  | 7859  |
| 7880  | 7884  | 7888  | 8010* | 8014* | 8021  | 8032* | 8050  | 8054  | 8058  | 8124* | 8128* | 8135  |
| 8146* | 8164  | 8168  | 8172  | 8254* | 8258* | 8268* | 8276* | 8294  | 8295  | 8314  | 8315  | 8338  |
| 8342  | 8346  | 8419* | 8423* | 8433* | 8441* | 8459  | 8460  | 8479  | 8480  | 8503  | 8507  | 8511  |
| 1458* | 2175  | 2176  | 2192  | 2193  | 2377  | 2477  | 2614  | 2714  |       |       |       |       |
| 1465* | 22369 | 2517  | 2606  | 2754  |       |       |       |       |       |       |       |       |
| 1466* | 2368  | 2522  | 2605  | 2759  |       |       |       |       |       |       |       |       |
| 1453* | 2380  | 2462  | 2617  | 2699  | 4023* | 4024* | 4028* | 4029* | 4224* | 4225* | 4229* | 4230* |
| 4423* | 4424* | 4428* | 4429* | 6283* | 6328  | 6329  | 6485* | 6530  | 6531  | 6694* | 6739  | 6740  |
| 6863* | 6908  | 6909  | 7607* | 7617* | 7621* | 7715  | 7716  | 7821* | 7831* | 7835* | 7929  | 7930  |
| 8018* | 8019* | 8132* | 8133* |       |       |       |       |       |       |       |       |       |
| 1463* | 2375  | 2487  | 2612  | 2724  | 7608* | 7616* | 7622* | 7732  | 7733  | 7822* | 7830* | 7836* |
| 7946  | 7947  |       |       |       |       |       |       |       |       |       |       |       |
| 1464* | 2370  | 2512  | 2507  | 2749  | 7609* | 7615* | 7623* | 7749  | 7750  | 7823* | 7829* | 7837* |
| 7963  | 7964  |       |       |       |       |       |       |       |       |       |       |       |
| 1455* | 2376  | 2482  | 2613  | 2719  |       |       |       |       |       |       |       |       |
| 1457* | 2379  | 2467  | 2616  | 2704  |       |       |       |       |       |       |       |       |
| 1460* | 2374  | 2492  | 2511  | 2729  | 2862* | 3041* |       |       |       |       |       |       |
| 1459* | 2279  | 2281  | 2371  | 2507  | 2608  | 2744  |       |       |       |       |       |       |
| 1448* | 2081  |       |       |       |       |       |       |       |       |       |       |       |
| 9411* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1213* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1346* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1349* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1347* |       |       |       |       |       |       |       |       |       |       |       |       |
| 1350* |       |       |       |       |       |       |       |       |       |       |       |       |

RPDT = 000026  
RPEC1 = 000044  
RPEC2 = 000046  
RPER1 = 000014

RPER2 = 000040

RPER3 = 000042

RPLA = 000020  
RPMR = 000024  
RPOF = 000032  
RPSN = 000030  
RPWC = 000002  
SAVREG = 104413  
SC = 100000  
SC1 = 000100  
SC10 = 001000  
SC2 = 000200  
SC20 = 002000









VV = 000100

|       |      |       |      |      |      |      |      |      |      |      |      |      |
|-------|------|-------|------|------|------|------|------|------|------|------|------|------|
| 1268# | 2419 | 2423  | 2444 | 2450 | 2656 | 2660 | 2681 | 2687 | 2871 | 2909 | 2913 | 2934 |
| 2940  | 3050 | 3084  | 3088 | 3092 | 3179 | 3183 | 3187 | 3324 | 3328 | 3332 | 3456 | 3460 |
| 3464  | 3536 | 3540  | 3544 | 3632 | 3673 | 3716 | 3739 | 3743 | 3747 | 3836 | 3877 | 3920 |
| 3943  | 3947 | 3951  | 4125 | 4129 | 4133 | 4326 | 4330 | 4334 | 4489 | 4493 | 4630 | 4653 |
| 4657  | 4661 | 4752  | 4775 | 4779 | 4783 | 4889 | 4956 | 4960 | 4964 | 5070 | 5137 | 5141 |
| 5145  | 5241 | 5287  | 5291 | 5295 | 5390 | 5436 | 5440 | 5444 | 5543 | 5595 | 5626 | 5630 |
| 5634  | 5732 | 5784  | 5815 | 5819 | 5823 | 5937 | 5960 | 5991 | 5995 | 5999 | 6043 | 6066 |
| 6097  | 6101 | 6105  | 6192 | 6196 | 6200 | 6369 | 6373 | 6571 | 6575 | 6767 | 6771 | 6775 |
| 6936  | 6940 | 6944  | 7065 | 7069 | 7073 | 7202 | 7206 | 7210 | 7362 | 7366 | 7370 | 7510 |
| 7514  | 7518 | 7637  | 7668 | 7672 | 7676 | 7851 | 7882 | 7886 | 7890 | 8052 | 8056 | 8060 |
| 8166  | 8170 | 8174  | 8340 | 8344 | 8348 | 8505 | 8509 | 8513 |      |      |      |      |
| 1931# | 2353 | 2415  | 2441 | 2444 | 2450 | 2590 | 2652 | 2678 | 2681 | 2687 | 2905 | 2931 |
| 2934  | 2940 | 3042# | 3084 | 3110 | 3113 | 3114 | 3119 | 3120 | 3179 | 3205 | 3208 | 3209 |
| 3214  | 3215 | 3324  | 3350 | 3353 | 3354 | 3359 | 3360 | 3456 | 3482 | 3485 | 3486 | 3491 |
| 3492  | 3536 | 3562  | 3565 | 3566 | 3571 | 3572 | 3632 | 3739 | 3765 | 3767 | 3770 | 3771 |
| 3776  | 3777 | 3836  | 3943 | 3969 | 3971 | 3974 | 3975 | 3980 | 3981 | 4125 | 4151 | 4153 |
| 4156  | 4157 | 4162  | 4163 | 4326 | 4352 | 4354 | 4357 | 4358 | 4363 | 4364 | 4511 | 4513 |
| 4516  | 4521 | 4630  | 4653 | 4679 | 4681 | 4684 | 4689 | 4752 | 4775 | 4801 | 4803 | 4806 |
| 4811  | 4889 | 4956  | 4982 | 4984 | 4987 | 4992 | 5070 | 5137 | 5163 | 5165 | 5168 | 5173 |
| 5241  | 5287 | 5313  | 5315 | 5318 | 5323 | 5390 | 5436 | 5462 | 5464 | 5467 | 5472 | 5543 |
| 5595  | 5626 | 5652  | 5654 | 5657 | 5662 | 5732 | 5784 | 5815 | 5841 | 5843 | 5846 | 5851 |
| 5960  | 5991 | 6017  | 6019 | 6022 | 6027 | 6066 | 6097 | 6123 | 6125 | 6128 | 6133 | 6192 |
| 6218  | 6223 | 6228  | 6391 | 6393 | 6396 | 6397 | 6402 | 6403 | 6593 | 6595 | 6598 | 6599 |
| 6604  | 6605 | 6767  | 6793 | 6795 | 6798 | 6803 | 6936 | 6962 | 6964 | 6967 | 6972 | 7065 |
| 7091  | 7096 | 7101  | 7202 | 7228 | 7233 | 7238 | 7362 | 7388 | 7390 | 7393 | 7398 | 7510 |
| 7536  | 7538 | 7541  | 7546 | 7637 | 7668 | 7694 | 7696 | 7699 | 7704 | 7851 | 7882 | 7908 |
| 7910  | 7913 | 7918  | 8052 | 8078 | 8080 | 8083 | 8084 | 8089 | 8090 | 8166 | 8192 | 8194 |
| 8197  | 8198 | 8203  | 8204 | 8340 | 8366 | 8368 | 8371 | 8376 | 8505 | 8531 | 8533 | 8536 |
| 8541  |      |       |      |      |      |      |      |      |      |      |      |      |

WAO = 000002  
WATCH = 001254

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1427# | 2329* | 2387  | 2566* | 2624  | 2802* | 2890  | 2981* | 3069  | 3155* | 3267  | 3300* | 3412  |
| 1569# | 6298  | 6491* | 6500  | 7035* | 7036  | 7043* | 7053  | 7172* | 7173  | 7180* | 7190  | 8288* |
| 6289# | 8453* | 8471  | 8636  | 8638* | 8640* |       |       |       |       |       |       |       |
| 8306  |       |       |       |       |       |       |       |       |       |       |       |       |

WCE = 040000  
WCF = 000040  
WCU = 000001  
WLF = 004000  
WRL = 004000  
WRU = 000400  
WSU = 000004  
\$AUTOB 001134  
\$BDAOR 001122

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1237# |       |       |       |       |       |       |       |       |       |       |       |       |
| 1286# |       |       |       |       |       |       |       |       |       |       |       |       |
| 1359# | 1377# |       |       |       |       |       |       |       |       |       |       |       |
| 1292# |       |       |       |       |       |       |       |       |       |       |       |       |
| 1273# |       |       |       |       |       |       |       |       |       |       |       |       |
| 1367# | 1385# |       |       |       |       |       |       |       |       |       |       |       |
| 1361# | 1379# |       |       |       |       |       |       |       |       |       |       |       |
| 1522# | 1988# | 9121  | 9269  |       |       |       |       |       |       |       |       |       |
| 1517# | 2131* | 2132* | 2152* | 2153* | 2176* | 2177* | 2193* | 2194* | 2214* | 2215* | 2229* | 2230* |
| 2246* | 2247* | 2261* | 2262* | 2342* | 2343* | 2413* | 2414* | 2461* | 2462* | 2466* | 2467* | 2471* |
| 2472* | 2476* | 2477* | 2481* | 2482* | 2486* | 2487* | 2491* | 2492* | 2496* | 2497* | 2501* | 2502* |
| 2506* | 2507* | 2511* | 2512* | 2516* | 2517* | 2521* | 2522* | 2579* | 2580* | 2650* | 2651* | 2698* |
| 2699* | 2703* | 2704* | 2708* | 2709* | 2713* | 2714* | 2718* | 2719* | 2723* | 2724* | 2728* | 2729* |
| 2733* | 2734* | 2738* | 2739* | 2743* | 2744* | 2748* | 2749* | 2753* | 2754* | 2758* | 2759* | 2817* |
| 2818* | 2829* | 2830* | 2844* | 2845* | 2869* | 2870* | 2903* | 2904* | 2996* | 2997* | 3008* | 3009* |
| 3023* | 3024* | 3048* | 3049* | 3082* | 3083* | 3177* | 3178* | 3226* | 3227* | 3243* | 3244* | 3322* |
| 3323* | 3371* | 3372* | 3388* | 3389* | 3454* | 3455* | 3534* | 3535* | 3621* | 3622* | 3654* | 3655* |
| 3671* | 3672* | 3697* | 3698* | 3714* | 3715* | 3737* | 3738* | 3825* | 3826* | 3858* | 3859* | 3875* |
| 3876* | 3901* | 3902* | 3918* | 3919* | 3941* | 3942* | 4041* | 4042* | 4061* | 4062* | 4098* | 4099* |
| 4123* | 4124* | 4175* | 4176* | 4242* | 4243* | 4262* | 4263* | 4299* | 4300* | 4324* | 4325* | 4376* |
| 4377* | 4441* | 4442* | 4461* | 4462* | 4483* | 4484* | 4540* | 4541* | 4557* | 4558* | 4619* | 4620* |
| 4651* | 4652* | 4741* | 4742* | 4773* | 4774* | 4890* | 4891* | 4912* | 4913* | 4929* | 4930* | 4954* |
| 4955* | 5071* | 5072* | 5093* | 5094* | 5110* | 5111* | 5135* | 5136* | 5230* | 5231* | 5258* | 5259* |

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 5285* | 5286* | 5379* | 5380* | 5407* | 5408* | 5434* | 5435* | 5532* | 5533* | 5565* | 5566* | 5573* |
| 5574* | 5596* | 5597* | 5624* | 5625* | 5721* | 5722* | 5754* | 5755* | 5762* | 5763* | 5785* | 5786* |
| 5813* | 5814* | 5935* | 5936* | 5961* | 5962* | 5989* | 5990* | 6041* | 6042* | 6067* | 6068* | 6095* |
| 6096* | 6190* | 6191* | 6310* | 6311* | 6329* | 6330* | 6343* | 6344* | 6363* | 6364* | 6415* | 6416* |
| 6512* | 6513* | 6531* | 6532* | 6545* | 6546* | 6565* | 6566* | 6617* | 6618* | 6698* | 6699* | 6728* |
| 6729* | 6740* | 6741* | 6765* | 6766* | 6867* | 6868* | 6897* | 6898* | 6909* | 6910* | 6934* | 6935* |
| 7063* | 7064* | 7200* | 7201* | 7314* | 7315* | 7335* | 7336* | 7360* | 7361* | 7462* | 7463* | 7483* |
| 7484* | 7508* | 7509* | 7638* | 7639* | 7666* | 7667* | 7716* | 7717* | 7724* | 7725* | 7733* | 7734* |
| 7741* | 7742* | 7750* | 7751* | 7758* | 7759* | 7852* | 7853* | 7880* | 7881* | 7930* | 7931* | 7938* |
| 7939* | 7947* | 7948* | 7955* | 7956* | 7964* | 7965* | 7972* | 7973* | 8023* | 8024* | 8050* | 8051* |
| 8137* | 8138* | 8164* | 8165* | 8295* | 8296* | 8315* | 8316* | 8338* | 8339* | 8460* | 8461* | 8480* |
| 8481* | 8503* | 8504* | 10065 | 10067 | 10069 | 10072 | 10074 | 10077 | 10082 |       |       |       |
| 1519* | 2130* | 2134  | 2138  | 2151* | 2155  | 2159  | 2175* | 2179  | 2183  | 2192* | 2196  | 2200  |
| 2213* | 2217  | 2221  | 2228* | 2232  | 2236  | 2245* | 2249  | 2253  | 2260* | 2264  | 2268  | 2281* |
| 2282  | 2341* | 2345  | 2352* | 2356  | 2430* | 2436* | 2442* | 2448* | 2463* | 2468* | 2473* | 2478* |
| 2483* | 2488* | 2493* | 2498* | 2503* | 2508* | 2513* | 2518* | 2523* | 2578* | 2582  | 2589* | 2593  |
| 2667* | 2673* | 2679* | 2685* | 2700* | 2705* | 2710* | 2715* | 2720* | 2725* | 2730* | 2735* | 2740* |
| 2745* | 2750* | 2755* | 2760* | 2816* | 2820  | 2828* | 2832  | 2836  | 2843* | 2847  | 2851  | 2868* |
| 2872  | 2876  | 2920* | 2926* | 2932* | 2938* | 2995* | 2999  | 3007* | 3011  | 3015  | 3022* | 3026  |
| 3030  | 3047* | 3051  | 3055  | 3099* | 3105* | 3111* | 3117* | 3194* | 3200* | 3206* | 3212* | 3225* |
| 3229  | 3233  | 3242* | 3246  | 3250  | 3339* | 3345* | 3351* | 3357* | 3370* | 3374  | 3378  | 3387* |
| 3391  | 3395  | 3471* | 3477* | 3483* | 3489* | 3551* | 3557* | 3563* | 3569* | 3620* | 3624  | 3631* |
| 3635  | 3653* | 3657  | 3661  | 3670* | 3674  | 3678  | 3696* | 3700  | 3704  | 3713* | 3717  | 3721  |
| 3754* | 3760* | 3768* | 3774* | 3824* | 3828  | 3835* | 3839  | 3857* | 3861  | 3865  | 3874* | 3878  |
| 3882  | 3900* | 3904  | 3908  | 3917* | 3921  | 3925  | 3958* | 3964* | 3972* | 3978* | 4040* | 4044  |
| 4048  | 4060* | 4064  | 4068  | 4097* | 4101  | 4105  | 4140* | 4146* | 4154* | 4160* | 4174* | 4178  |
| 4182  | 4241* | 4245  | 4249  | 4261* | 4265  | 4269  | 4298* | 4302  | 4306  | 4341* | 4347* | 4355* |
| 4361* | 4375* | 4379  | 4383  | 4440* | 4444  | 4448  | 4460* | 4464  | 4468  | 4500* | 4506* | 4514* |
| 4519* | 4539* | 4543  | 4547  | 4556* | 4560  | 4564  | 4618* | 4622  | 4629* | 4633  | 4668* | 4674* |
| 4682* | 4687* | 4740* | 4744  | 4751* | 4755  | 4790* | 4796* | 4804* | 4809* | 4897* | 4902* | 4904  |
| 4911* | 4915  | 4919  | 4928* | 4932  | 4936  | 4971* | 4977* | 4985* | 4990* | 5078* | 5083* | 5085  |
| 5092* | 5096  | 5100  | 5109* | 5113  | 5117  | 5152* | 5158* | 5166* | 5171* | 5229* | 5233  | 5240* |
| 5244  | 5257* | 5261  | 5265  | 5302* | 5308* | 5316* | 5321* | 5378* | 5382  | 5389* | 5393  | 5406* |
| 5410  | 5414  | 5451* | 5457* | 5465* | 5470* | 5531* | 5535  | 5542* | 5546  | 5564* | 5568  | 5572* |
| 5575  | 5603* | 5608* | 5610  | 5641* | 5647* | 5655* | 5660* | 5720* | 5724  | 5731* | 5735  | 5753* |
| 5757  | 5761* | 5764  | 5792* | 5797* | 5799  | 5830* | 5836* | 5844* | 5849* | 5934* | 5938  | 5942  |
| 5968* | 5973* | 5975  | 6006* | 6012* | 6020* | 6025* | 6040* | 6044  | 6048  | 6074* | 6079* | 6081  |
| 6112* | 6118* | 6126* | 6131* | 6207* | 6213* | 6221* | 6226* | 6309* | 6313  | 6317  | 6328* | 6332  |
| 6342* | 6346  | 6350  | 6380* | 6386* | 6394* | 6400* | 6414* | 6418  | 6422  | 6511* | 6515  | 6519  |
| 6530* | 6534  | 6544* | 6548  | 6552  | 6582* | 6588* | 6596* | 6602* | 6616* | 6620  | 6624  | 6697* |
| 6701  | 6705  | 6727* | 6731  | 6739* | 6743  | 6782* | 6788* | 6796* | 6801* | 6866* | 6870  | 6874  |
| 6896* | 6900  | 6908* | 6912  | 6951* | 6957* | 6965* | 6970* | 7080* | 7086* | 7094* | 7099* | 7217* |
| 7223* | 7231* | 7236* | 7313* | 7317  | 7321  | 7334* | 7338  | 7342  | 7377* | 7383* | 7391* | 7396* |
| 7461* | 7465  | 7469  | 7482* | 7486  | 7490  | 7525* | 7531* | 7539* | 7544* | 7645* | 7650* | 7652  |
| 7683* | 7689* | 7697* | 7702* | 7715* | 7719  | 7723* | 7726  | 7732* | 7736  | 7740* | 7743  | 7749* |
| 7753  | 7757* | 7760  | 7859* | 7864* | 7866  | 7897* | 7903* | 7911* | 7916* | 7929* | 7933  | 7937* |
| 7940  | 7946* | 7950  | 7954* | 7957  | 7963* | 7967  | 7971* | 7974  | 8033* | 8034  | 8067* | 8073* |
| 8081* | 8087* | 8147* | 8148  | 8181* | 8187* | 8195* | 8201* | 8294* | 8298  | 8314* | 8318  | 8355* |
| 8361* | 8369* | 8374* | 8459* | 8463  | 8479* | 8483  | 8520* | 8526* | 8534* | 8539* | 10065 | 10067 |
| 10069 | 10072 | 10074 | 10077 | 10082 |       |       |       |       |       |       |       |       |
| 1545* | 8741  | 8761  | 9074  | 9263  |       |       |       |       |       |       |       |       |
| 8853* | 8863* | 8870  | 8879* | 8884* |       |       |       |       |       |       |       |       |
| 9107* | 9407  |       |       |       |       |       |       |       |       |       |       |       |
| 1505* | 1942  | 1943  | 1951  | 1955  | 1956  | 1957  |       |       |       |       |       |       |
| 1537* | 1538* |       |       |       |       |       |       |       |       |       |       |       |
| 1537* | 1538* |       |       |       |       |       |       |       |       |       |       |       |

\$BODAT 001126

\$BELL 001202  
\$CHARC 057644  
\$CKSWR 060574  
\$CMTAG 001100  
\$CM1 = 000001  
\$CM2 = 000002





SSVLAD 057056  
SSVPC = 000200  
SSWR = 166000

|       |       |      |      |      |      |      |      |      |      |       |      |      |  |
|-------|-------|------|------|------|------|------|------|------|------|-------|------|------|--|
| 8686  | 8711* |      |      |      |      |      |      |      |      |       |      |      |  |
| 1482* | 1487  |      |      |      |      |      |      |      |      |       |      |      |  |
| 1063* | 1074  | 1079 | 1080 | 1081 | 1082 | 1083 | 1084 | 1085 | 1543 | 1544  | 1545 | 1956 |  |
| 1957  | 1959  | 1960 | 2120 | 2318 | 2555 | 2793 | 2972 | 3148 | 3293 | 3436  | 3516 | 3607 |  |
| 3811  | 4016  | 4217 | 4416 | 4594 | 4716 | 4846 | 5027 | 5205 | 5354 | 5507  | 5696 | 5882 |  |
| 6161  | 6258  | 6460 | 6669 | 6838 | 7004 | 7141 | 7281 | 7429 | 7587 | 7801  | 8004 | 8118 |  |
| 8248  | 8413  | 8555 | 8563 | 8590 | 8596 | 8598 | 8668 | 8669 | 8670 | 8671  | 8672 | 8677 |  |
| 8689  | 8691  | 8692 | 8693 | 8700 | 8701 | 8702 | 8713 | 8716 | 8719 | 8720* | 8726 | 8727 |  |
| 8728  | 8729  | 8730 | 8739 | 8746 | 8751 | 8755 | 8761 |      |      |       |      |      |  |
| 8672  |       |      |      |      |      |      |      |      |      |       |      |      |  |

SSWRMK= 000000  
\$TIMES 001176

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1543* | 1956* | 2120* | 2318* | 2555* | 2793* | 2972* | 3148* | 3293* | 3436* | 3516* | 3607* | 3811* |
| 4016* | 4217* | 4416* | 4594* | 4716* | 4846* | 5027* | 5205* | 5354* | 5507* | 5696* | 5882* | 6161* |
| 6258* | 6460* | 6669* | 6838* | 7004* | 7141* | 7281* | 7429* | 7587* | 7801* | 8004* | 8118* | 8248* |
| 8413* | 8563* | 8700* | 8707  | 8710* | 8719  |       |       |       |       |       |       |       |

\$TKB 001146  
\$TKCNT 060322  
\$TKINT 060332  
\$TKQEN= 060331  
\$TKQIN 060324  
\$TKQOU 060326  
\$TKQSR 060330  
\$TKS 001144  
\$TKSRV 060402  
\$TMPO 001164

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1528* | 9034  | 9055  | 9064  | 9083  | 9111  | 9138  |       |       |       |       |       |       |
| 9035* | 9050* | 9072  | 9089* | 9195  | 9197* |       |       |       |       |       |       |       |
| 1980  | 2041  | 9050* | 9124  |       |       |       |       |       |       |       |       |       |
| 9039* | 9097  | 9200  |       |       |       |       |       |       |       |       |       |       |
| 9036* | 9051* | 9052  | 9095* | 9096* | 9097  | 9099* |       |       |       |       |       |       |
| 9037* | 9052* | 9198  | 9199* | 9200  | 9202* |       |       |       |       |       |       |       |
| 9038* | 9051  | 9099  | 9202  |       |       |       |       |       |       |       |       |       |
| 1527* | 9034  | 9056* | 9079* | 9081  | 9087* | 9109  | 9125* | 9135  | 9159* |       |       |       |
| 9053  | 9064* |       |       |       |       |       |       |       |       |       |       |       |
| 1538* | 2134* | 2135* | 2136  | 2155* | 2156* | 2157  | 2179* | 2180* | 2181  | 2196* | 2197* | 2198  |
| 2217* | 2218* | 2219  | 2232* | 2233* | 2234  | 2249* | 2250* | 2251  | 2264* | 2265* | 2266  | 2356* |
| 2357* | 2358  | 2418* | 2419* | 2424  | 2426  | 2433  | 2593* | 2594* | 2595  | 2655* | 2656* | 2661  |
| 2663  | 2670  | 2832* | 2833* | 2834  | 2847* | 2849  | 2872* | 2873* | 2874  | 2908* | 2909* | 2909* |
| 2914  | 2916  | 2923  | 3011* | 3012* | 3013  | 3026* | 3027* | 3028  | 3051* | 3052* | 3053  | 3087* |
| 3088* | 3093  | 3095  | 3102  | 3182* | 3183* | 3188  | 3190  | 3197  | 3229* | 3230* | 3231  | 3246* |
| 3247* | 3248  | 3327* | 3328* | 3333  | 3335  | 3342  | 3374* | 3375* | 3376  | 3391* | 3392* | 3393  |
| 3459* | 3460* | 3465  | 3467  | 3474  | 3539* | 3540* | 3545  | 3547  | 3554  | 3635* | 3636* | 3637  |
| 3657* | 3658* | 3659  | 3674* | 3675* | 3676  | 3700* | 3701* | 3702  | 3717* | 3718* | 3719  | 3742* |
| 3743* | 3748  | 3750  | 3757  | 3839* | 3840* | 3841  | 3861* | 3862* | 3863  | 3878* | 3879* | 3880  |
| 3904* | 3905* | 3906  | 3921* | 3922* | 3923  | 3946* | 3947* | 3952  | 3954  | 3961  | 4044* | 4045* |
| 4046  | 4064* | 4065* | 4066  | 4101* | 4102* | 4103  | 4128* | 4129* | 4134  | 4136  | 4143  | 4178* |
| 4179* | 4180  | 4245* | 4246* | 4247  | 4265* | 4266* | 4267  | 4302* | 4303* | 4304  | 4329* | 4330* |
| 4335  | 4337  | 4344  | 4379* | 4380* | 4381  | 4444* | 4445* | 4446  | 4464* | 4465* | 4466  | 4488* |
| 4489* | 4494  | 4496  | 4503  | 4543* | 4544* | 4545  | 4560* | 4561* | 4562  | 4633* | 4634* | 4635  |
| 4656* | 4657* | 4662  | 4664  | 4671  | 4755* | 4756* | 4757  | 4778* | 4779* | 4784  | 4786  | 4793  |
| 4894* | 4899  | 4902  | 4915* | 4916* | 4917  | 4932* | 4933* | 4934  | 4959* | 4960* | 4965  | 4967  |
| 4974  | 5075* | 5080  | 5083  | 5096* | 5097* | 5098  | 5113* | 5114* | 5115  | 5140* | 5141* | 5146  |
| 5148  | 5155  | 5244* | 5245* | 5246  | 5261* | 5262* | 5263  | 5290* | 5291* | 5296  | 5298  | 5305  |
| 5393* | 5394* | 5395  | 5410* | 5411* | 5412  | 5439* | 5440* | 5445  | 5447  | 5454  | 5546* | 5547* |
| 5548  | 5600* | 5605  | 5608  | 5629* | 5630* | 5635  | 5637  | 5644  | 5735* | 5736* | 5737  | 5789* |
| 5794  | 5797  | 5818* | 5819* | 5824  | 5826  | 5833  | 5938* | 5939* | 5940  | 5965* | 5970  | 5973  |
| 5994* | 5995* | 6000  | 6002  | 6009  | 6044* | 6045* | 6046  | 6071* | 6076  | 6079  | 6100* | 6101* |
| 6106  | 6108  | 6115  | 6179* | 6180* | 6181  | 6195* | 6196* | 6201  | 6203  | 6210  | 6313* | 6314* |
| 6315  | 6346* | 6347* | 6348  | 6368* | 6369* | 6374  | 6376  | 6383  | 6418* | 6419* | 6420  | 6515* |
| 6516* | 6517  | 6548* | 6549* | 6550  | 6570* | 6571* | 6576  | 6578  | 6585  | 6620* | 6621* | 6622  |
| 6701* | 6702* | 6703  | 6770* | 6771* | 6776  | 6778  | 6785  | 6870* | 6871* | 6872  | 6939* | 6940* |
| 6945  | 6947  | 6954  | 7068* | 7069* | 7074  | 7076  | 7083  | 7205* | 7206* | 7211  | 7213  | 7220  |
| 7317* | 7318* | 7319  | 7338* | 7339* | 7340  | 7365* | 7366* | 7371  | 7373  | 7380  | 7465* | 7466* |
| 7467  | 7486* | 7487* | 7488  | 7513* | 7514* | 7519  | 7521  | 7528  | 7642* | 7647  | 7650  | 7671* |
| 7672* | 7677  | 7679  | 7686  | 7856* | 7861  | 7864  | 7885* | 7886* | 7891  | 7893  | 7900  | 8034* |
| 8035* | 8036  | 8055* | 8056* | 8061  | 8063  | 8070  | 8148* | 8149* | 8150  | 8169* | 8170* | 8175  |
| 8177  | 8184  | 8343* | 8344* | 8349  | 8351  | 8358  | 8509* | 8509* | 8514  | 8516  | 8523  |       |

STMP1 001166

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1539* | 2354* | 2355* | 2357  | 2422* | 2423* | 2424  | 2438  | 2591* | 2592* | 2594  | 2659* | 2660* |
| 2661  | 2675  | 2912* | 2913* | 2914  | 2928  | 3091* | 3092* | 3093  | 3107  | 3186* | 3187* | 3188  |
| 3202  | 3331* | 3332* | 3333  | 3347  | 3463* | 3464* | 3465  | 3479  | 3543* | 3544* | 3545  | 3559  |
| 3633* | 3634* | 3636  | 3746* | 3747* | 3748  | 3762  | 3837* | 3838* | 3840  | 3950* | 3951* | 3952  |
| 3966  | 4132* | 4133* | 4134  | 4148  | 4333* | 4334* | 4335  | 4349  | 4492* | 4493* | 4494  | 4508  |
| 4631* | 4632* | 4634  | 4660* | 4661* | 4662  | 4676  | 4753* | 4754* | 4756  | 4782* | 4783* | 4784  |
| 4798  | 4963* | 4964* | 4965  | 4979  | 5144* | 5145* | 5146  | 5160  | 5242* | 5243* | 5245  | 5294* |
| 5295* | 5296  | 5310  | 5391* | 5392* | 5394  | 5443* | 5444* | 5445  | 5459  | 5544* | 5545* | 5547  |
| 5633* | 5634* | 5635  | 5649  | 5733* | 5734* | 5736  | 5822* | 5823* | 5824  | 5838  | 5998* | 5999* |
| 6000  | 6014  | 6104* | 6105* | 6106  | 6120  | 6199* | 6200* | 6201  | 6215  | 6372* | 6373* | 6374  |
| 6388  | 6574* | 6575* | 6576  | 6590  | 6774* | 6775* | 6776  | 6790  | 6943* | 6944* | 6945  | 6959  |
| 7072* | 7073* | 7074  | 7088  | 7209* | 7210* | 7211  | 7225  | 7369* | 7370* | 7371  | 7385  | 7517* |
| 7518* | 7519  | 7533  | 7675* | 7676* | 7677  | 7691  | 7889* | 7890* | 7891  | 7905  | 8026* | 8027* |
| 8035  | 8059* | 8060* | 8061  | 8075  | 8140* | 8141* | 8149  | 8173* | 8174* | 8175  | 8189  | 8347* |

STMP2 001170

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1540* | 2417* | 2418  | 2430  | 2442* | 2444* | 2445  | 2654* | 2655  | 2667  | 2679  | 2681* | 2682  |
| 2907* | 2908  | 2920  | 2932  | 2934* | 2935  | 3086* | 3087  | 3099  | 3111  | 3113* | 3114  | 3181* |
| 3182  | 3194  | 3206  | 3208* | 3209  | 3326* | 3327  | 3339  | 3351  | 3353* | 3354  | 3458* | 3459  |
| 3471  | 3483  | 3485* | 3486  | 3538* | 3539  | 3551  | 3563  | 3565* | 3566  | 3741* | 3742  | 3754  |
| 3768  | 3770* | 3771  | 3945* | 3946  | 3958  | 3972  | 3974* | 3975  | 4127* | 4128  | 4140  | 4154  |
| 4156* | 4157  | 4328* | 4329  | 4341  | 4355  | 4357* | 4358  | 4487* | 4488  | 4500  | 4514  | 4516  |
| 4655* | 4656  | 4668  | 4682  | 4684  | 4777* | 4778  | 4790  | 4804  | 4806  | 4958* | 4959  | 4971  |
| 4985  | 4987  | 5139* | 5140  | 5152  | 5166  | 5168  | 5289* | 5290  | 5302  | 5316  | 5318  | 5438* |
| 5439  | 5451  | 5465  | 5467  | 5628* | 5629  | 5641  | 5655  | 5657  | 5817* | 5818  | 5830  | 5844  |
| 5846  | 5993* | 5994  | 6006  | 6020  | 6022  | 6099* | 6100  | 6112  | 6126  | 6128  | 6194* | 6195  |
| 6207  | 6221  | 6223  | 6367* | 6368  | 6380  | 6394  | 6396* | 6397  | 6569* | 6570  | 6582  | 6596  |
| 6598* | 6599  | 6769* | 6770  | 6782  | 6796  | 6798  | 6938* | 6939  | 6951  | 6965  | 6967  | 7067* |
| 7068  | 7080  | 7094  | 7096  | 7204* | 7205  | 7217  | 7231  | 7233  | 7364* | 7365  | 7377  | 7391  |
| 7393  | 7512* | 7513  | 7525  | 7539  | 7541  | 7670* | 7671  | 7683  | 7697  | 7699  | 7884* | 7885  |
| 7897  | 7911  | 7913  | 8054* | 8055  | 8067  | 8081  | 8083* | 8084  | 8168* | 8169  | 8181  | 8195  |

STMP3 001172

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 8197* | 8198  | 8342* | 8343  | 8355  | 8369  | 8371  | 8507* | 8508  | 8520  | 8534  | 8536  | 10091 |
| 1541* | 2421* | 2422  | 2436  | 2448  | 2450* | 2451  | 2658* | 2659  | 2673  | 2685  | 2687* | 2688  |
| 2911* | 2912  | 2926  | 2938  | 2940* | 2941  | 3090* | 3091  | 3105  | 3117  | 3119* | 3120  | 3185* |
| 3186  | 3200  | 3212  | 3214* | 3215  | 3330* | 3331  | 3345  | 3357  | 3359* | 3360  | 3462* | 3463  |
| 3477  | 3489  | 3491* | 3492  | 3542* | 3543  | 3557  | 3569  | 3571* | 3572  | 3745* | 3746  | 3760  |
| 3774  | 3776* | 3777  | 3949* | 3950  | 3964  | 3978  | 3980* | 3981  | 4131* | 4132  | 4146  | 4160  |
| 4162* | 4163  | 4332* | 4333  | 4347  | 4361  | 4363* | 4364  | 4491* | 4492  | 4506  | 4519  | 4521  |
| 4659* | 4660  | 4674  | 4687  | 4689  | 4781* | 4782  | 4796  | 4809  | 4811  | 4962* | 4963  | 4977  |
| 4990  | 4992  | 5143* | 5144  | 5158  | 5171  | 5173  | 5293* | 5294  | 5308  | 5321  | 5323  | 5442* |
| 5443  | 5457  | 5470  | 5472  | 5632* | 5633  | 5647  | 5660  | 5662  | 5821* | 5822  | 5836  | 5849  |
| 5851  | 5997* | 5998  | 6012  | 6025  | 6027  | 6103* | 6104  | 6118  | 6131  | 6133  | 6198* | 6199  |
| 6213  | 6226  | 6228  | 6371* | 6372  | 6386  | 6400  | 6402* | 6403  | 6573* | 6574  | 6588  | 6602  |
| 6604* | 6605  | 6773* | 6774  | 6788  | 6801  | 6803  | 6942* | 6943  | 6957  | 6970  | 6972  | 7071* |
| 7072  | 7086  | 7099  | 7101  | 7208* | 7209  | 7223  | 7236  | 7238  | 7368* | 7369  | 7383  | 7396  |
| 7398  | 7516* | 7517  | 7531  | 7544  | 7546  | 7674* | 7675  | 7689  | 7702  | 7704  | 7888* | 7889  |
| 7903  | 7916  | 7918  | 8058* | 8059  | 8073  | 8087  | 8089* | 8090  | 8172* | 8173  | 8187  | 8201  |

STMP4 001174

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 8203* | 8204  | 8346* | 8347  | 8361  | 8374  | 8376  | 8511* | 8512  | 8526  | 8539  | 8541  | 10091 |
| 1542* | 2138* | 2139* | 2140  | 2159* | 2160* | 2161  | 2183* | 2184* | 2185  | 2200* | 2201* | 2202  |
| 2221* | 2222* | 2223  | 2236* | 2237  | 2238  | 2253* | 2254* | 2255  | 2268* | 2269* | 2270  | 2836* |
| 2837* | 2838  | 2851* | 2852* | 2853  | 2876* | 2877* | 2878  | 3015* | 3016* | 3017  | 3030* | 3031* |
| 3032  | 3055* | 3056* | 3057  | 3233* | 3234* | 3235  | 3250* | 3251* | 3252  | 3378* | 3379* | 3380  |
| 3395* | 3396* | 3397  | 3661* | 3662* | 3663  | 3678* | 3679* | 3680  | 3704* | 3705* | 3706  | 3721* |
| 3722* | 3723  | 3865* | 3866* | 3867  | 3882* | 3883* | 3884  | 3908* | 3909* | 3910  | 3925* | 3926* |
| 3927  | 4048* | 4049* | 4050  | 4068* | 4069* | 4070  | 4105* | 4106* | 4107  | 4182* | 4183* | 4194  |
| 4249* | 4250* | 4251  | 4269* | 4270* | 4271  | 4306* | 4307* | 4308  | 4383* | 4384* | 4385  | 4448* |
| 4449* | 4450  | 4468* | 4469* | 4470  | 4547* | 4548* | 4549  | 4564* | 4565* | 4566  | 4919* | 4920* |

\$TN = 000047

\$TPB 001152  
\$TPFLG 001157  
\$TPS 001150  
\$TRAP 061722  
\$TRAP2 061744  
\$TRP = 000015

\$TRPAD 061756  
\$TSTNM 001102

\$TTYIN 061422  
\$TYPBN= \*\*\*\*\* U  
\$TYPDS 060076  
\$TYPE 057430  
\$TYPEC 057600  
\$TYPEX 057646  
\$TYPOC 057674  
\$TYPON 057710  
\$TYPOS 057650  
\$XTSTR 056702  
\$\$GET4= 000000  
\$OFILL 060073  
\$4OCAT= \*\*\*\*\* U  
= 070730

|       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 4921  | 4936* | 4937* | 4938  | 5100* | 5101* | 5102  | 5117* | 5118* | 5119  | 5265* | 5266* | 5267  |
| 5414* | 5415* | 5416  | 5942* | 5943* | 5944  | 6048* | 6049* | 6050  | 6317* | 6318* | 6319  | 6350* |
| 6351* | 6352  | 6422* | 6423* | 6424  | 6519* | 6520* | 6521  | 6552* | 6553* | 6554  | 6624* | 6625* |
| 6626  | 6705* | 6706* | 6707  | 6874* | 6875* | 6876  | 7321* | 7322* | 7323  | 7342* | 7343* | 7344  |
| 7469* | 7470* | 7471  | 7490* | 7491* | 7492  |       |       |       |       |       |       |       |
| 1063# | 1074  | 2098  | 2111  | 2120# | 2121  | 2291  | 2309  | 2318# | 2319  | 2528  | 2546  | 2555# |
| 2556  | 2766  | 2784  | 2793# | 2794  | 2947  | 2963  | 2972# | 2973  | 3127  | 3139  | 3148# | 3149  |
| 3272  | 3284  | 3293# | 3294  | 3418  | 3427  | 3436# | 3437  | 3498  | 3507  | 3516# | 3517  | 3579  |
| 3598  | 3607# | 3608  | 3783  | 3802  | 3811# | 3812  | 3988  | 4007  | 4016# | 4017  | 4190  | 4208  |
| 4217# | 4218  | 4393  | 4407  | 4416# | 4417  | 4573  | 4585  | 4594# | 4595  | 4695  | 4707  | 4716# |
| 4717  | 4818  | 4837  | 4846# | 4847  | 4998  | 5018  | 5027# | 5028  | 5180  | 5196  | 5205# | 5206  |
| 5329  | 5345  | 5354# | 5355  | 5479  | 5498  | 5507# | 5508  | 5668  | 5687  | 5696# | 5697  | 5858  |
| 5873  | 5882# | 5883  | 6140  | 6152  | 6161# | 6162  | 6235  | 6249  | 6258# | 6259  | 6437  | 6451  |
| 6460# | 6461  | 6640  | 6660  | 6669# | 6670  | 6809  | 6829  | 6838# | 6839  | 6979  | 6995  | 7004# |
| 7005  | 7116  | 7132  | 7141# | 7142  | 7255  | 7272  | 7281# | 7282  | 7404  | 7420  | 7429# | 7430  |
| 7553  | 7578  | 7587# | 7588  | 7589  | 7709  | 7767  | 7792  | 7801# | 7802  | 7803  | 7923  | 7982  |
| 7995  | 8004# | 8005  | 8096  | 8109  | 8118# | 8119  | 8218  | 8239  | 8248# | 8249  | 8383  | 8404  |
| 8413# | 8414  | 10175 |       |       |       |       |       |       |       |       |       |       |
| 1530# | 8876* | 8887  |       |       |       |       |       |       |       |       |       |       |
| 1534# | 8834  | 8887  |       |       |       |       |       |       |       |       |       |       |
| 1529# | 8874  | 8887  |       |       |       |       |       |       |       |       |       |       |
| 1953  | 9376# |       |       |       |       |       |       |       |       |       |       |       |
| 9387# | 9398  | 9401# | 9402# | 9403# | 9404# | 9405  | 9406# | 9407  | 9408# | 9409# | 9410# | 9411# |
| 9391# | 9400# |       |       |       |       |       |       |       |       |       |       |       |
| 9412# | 9413# |       |       |       |       |       |       |       |       |       |       |       |
| 9381  | 9398# |       |       |       |       |       |       |       |       |       |       |       |
| 1507# | 2117* | 2315* | 2552* | 2790* | 2969* | 3145* | 3290* | 3433* | 3513* | 3604* | 3808* | 4013* |
| 4214* | 4413* | 4591* | 4713* | 4843* | 5024* | 5202* | 5351* | 5504* | 5693* | 5879* | 6158* | 6255* |
| 6457* | 6666* | 6835* | 7001* | 7138* | 7278* | 7426* | 7584* | 7798* | 8001* | 8115* | 8245* | 8410* |
| 8562* | 8667  | 8711* | 8716  | 8720  | 8735  | 8738  | 8761  |       |       |       |       |       |
| 9213  | 9214  | 9226  | 9244  | 9258  | 9262# |       |       |       |       |       |       |       |
| 9404  |       |       |       |       |       |       |       |       |       |       |       |       |
| 8976# | 9403  |       |       |       |       |       |       |       |       |       |       |       |
| 8834# | 9391  | 9399  |       |       |       |       |       |       |       |       |       |       |
| 8855  | 8862  | 8869  | 8874# | 8875  | 9161  |       |       |       |       |       |       |       |
| 8880  | 8882  | 8885# |       |       |       |       |       |       |       |       |       |       |
| 8917# | 9400  |       |       |       |       |       |       |       |       |       |       |       |
| 8916  | 8919# | 9402  |       |       |       |       |       |       |       |       |       |       |
| 8912# | 9401  |       |       |       |       |       |       |       |       |       |       |       |
| 8680# |       |       |       |       |       |       |       |       |       |       |       |       |
| 8590# |       |       |       |       |       |       |       |       |       |       |       |       |
| 8913* | 8917* | 8927  | 8962# |       |       |       |       |       |       |       |       |       |
| 8677  | 8748  |       |       |       |       |       |       |       |       |       |       |       |
| 1470# | 1474# | 1482  | 1483# | 1485# | 1487# | 1504# | 1549  | 1946  | 1959  | 1960  | 2045  | 2046  |
| 2145  | 2166  | 2464  | 2469  | 2474  | 2479  | 2484  | 2489  | 2494  | 2499  | 2504  | 2509  | 2514  |
| 2519  | 2524  | 2701  | 2706  | 2711  | 2716  | 2721  | 2726  | 2731  | 2736  | 2741  | 2746  | 2751  |
| 2756  | 2761  | 3220  | 3365  | 4055  | 4075  | 4256  | 4276  | 4455  | 4475  | 4526  | 5581  | 5770  |
| 5924  | 6712  | 6881  | 7112  | 7249  | 7307  | 7455  | 8304  | 8307  | 8324  | 8469  | 8472  | 8489  |
| 8560  | 8575# | 8598  | 8599# | 8719  | 8720  | 8761  | 8816# | 8887  | 9030# | 9034  | 9038# | 9039  |
| 9040# | 9262# | 9263  | 9269# | 9323  |       |       |       |       |       |       |       |       |





|        |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| MMI    | 4018# | 4219  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| MORETA | 1497# | 1550  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| MSG    | 2095# | 2100  | 2290# | 2293  | 2527# | 2530  | 2765# | 2768  | 2946# | 2949  | 3126# | 3129  | 3271# | 3274  | 3417# |
|        | 3420  | 3497# | 3500  | 3578# | 3581  | 3782# | 3785  | 3987# | 3990  | 4189# | 4192  | 4392# | 4395  | 4572# | 4575  |
|        | 4694# | 4697  | 4817# | 4820  | 4997# | 5000  | 5179# | 5182  | 5328# | 5331  | 5478# | 5481  | 5667# | 5670  | 5857# |
|        | 5860  | 6139# | 6142  | 6234# | 6237  | 6436# | 6439  | 6639# | 6642  | 6808# | 6811  | 6979# | 6981  | 7115# | 7118  |
|        | 7254# | 7257  | 7403# | 7406  | 7552# | 7555  | 7766# | 7769  | 7981# | 7984  | 8095# | 8098  | 8217# | 8220  | 8382# |
|        | 8385  |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| MULT   | 1197# |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| NEUTRA | 1063# | 2409  | 2646  | 2898  | 3077  | 3173  | 3318  | 3450  | 3530  | 3733  | 3937  | 4119  | 4320  | 4479  | 4647  |
|        | 4769  | 4950  | 5131  | 5281  | 5430  | 5620  | 5809  | 5985  | 6091  | 6186  | 6359  | 6561  | 6761  | 6930  | 7059  |
|        | 7196  | 7356  | 7504  | 7662  | 7876  | 8046  | 8160  | 8334  | 8499  |       |       |       |       |       |       |
| NEWTST | 1197# | 2098  | 2291  | 2528  | 2766  | 2947  | 3127  | 3272  | 3418  | 3498  | 3579  | 3783  | 3988  | 4190  | 4393  |
|        | 4573  | 4695  | 4818  | 4998  | 5180  | 5329  | 5479  | 5668  | 5858  | 6140  | 6235  | 6437  | 6640  | 6809  | 6979  |
|        | 7116  | 7255  | 7404  | 7553  | 7767  | 7982  | 8096  | 8218  | 8383  |       |       |       |       |       |       |
| NN     | 8250# | 8415  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| OO     | 3609# | 3813  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| POP    | 1197# | 9017  | 9312  | 9361  |       |       |       |       |       |       |       |       |       |       |       |
| PUSH   | 1197# | 8976  | 9286  | 9341  |       |       |       |       |       |       |       |       |       |       |       |
| RELEAS | 1063# | 3167  | 3312  | 3727  | 3931  | 4113  | 4314  | 4641  | 4763  | 4879  | 4944  | 5060  | 5125  | 5275  | 5424  |
|        | 5585  | 5614  | 5774  | 5803  | 5950  | 5979  | 6056  | 6085  | 6755  | 6924  | 7350  | 7498  | 7627  | 7656  | 7841  |
|        | 7970  | 8039  | 8153  | 8328  | 8493  |       |       |       |       |       |       |       |       |       |       |
| REPORT | 1197# |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| RR     | 8006# | 8120  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| SCOPE  | 1092# | 2288  | 2526  | 2763  | 2945  | 3124  | 3270  | 3415  | 3496  | 3576  | 3781  | 3985  | 4188  | 4389  | 4570  |
|        | 4693  | 4815  | 4996  | 5177  | 5327  | 5476  | 5666  | 5855  | 6137  | 6232  | 6435  | 6637  | 6807  | 6976  | 7114  |
|        | 7251  | 7402  | 7550  | 7708  | 7922  | 8094  | 8208  | 8381  | 8546  |       |       |       |       |       |       |
| SEIZE  | 1063# | 2332  | 2569  | 3158  | 3303  | 3611  | 3815  | 4079  | 4280  | 4609  | 4731  | 4861  | 5042  | 5220  | 5369  |
|        | 5522  | 5711  | 6272  | 6474  | 6683  | 6852  | 7019  | 7156  | 7600  | 7814  | 8263  | 8428  |       |       |       |
| SELECT | 1063# | 2126  | 2147  | 2172  | 2189  | 2210  | 2242  | 2338  | 2349  | 2366  | 2575  | 2586  | 2603  | 2795  | 2813  |
|        | 2825  | 2882  | 2974  | 2992  | 3004  | 3061  | 3170  | 3222  | 3234  | 3262  | 3315  | 3367  | 3384  | 3407  | 3438  |
|        | 3518  | 3617  | 3628  | 3650  | 3667  | 3693  | 3710  | 3730  | 3821  | 3832  | 3854  | 3871  | 3897  | 3914  | 3934  |
|        | 4037  | 4057  | 4116  | 4171  | 4238  | 4258  | 4317  | 4372  | 4437  | 4457  | 4536  | 4553  | 4615  | 4626  | 4644  |
|        | 4737  | 4748  | 4766  | 4866  | 4882  | 4892  | 4895  | 4908  | 4925  | 4947  | 5049  | 5063  | 5073  | 5076  | 5089  |
|        | 5106  | 5128  | 5226  | 5237  | 5250  | 5278  | 5375  | 5386  | 5399  | 5427  | 5528  | 5539  | 5556  | 5588  | 5596  |
|        | 5601  | 5617  | 5717  | 5728  | 5745  | 5777  | 5787  | 5790  | 5806  | 5906  | 5914  | 5920  | 5953  | 5963  | 5966  |
|        | 5982  | 6036  | 6059  | 6069  | 6072  | 6088  | 6290  | 6301  | 6411  | 6492  | 6503  | 6613  | 6715  | 6723  | 6736  |
|        | 6758  | 6884  | 6892  | 6905  | 6927  | 7049  | 7186  | 7294  | 7327  | 7353  | 7442  | 7475  | 7501  | 7610  | 7613  |
|        | 7619  | 7624  | 7630  | 7640  | 7643  | 7659  | 7824  | 7827  | 7833  | 7838  | 7844  | 7854  | 7857  | 7873  | 8029  |
|        | 8043  | 8143  | 8157  | 8270  | 8277  | 8331  | 8435  | 8442  | 8496  |       |       |       |       |       |       |
| SETATA | 1063# | 4019  | 4220  | 4419  |       |       |       |       |       |       |       |       |       |       |       |
| SETPRI | 1197# | 9191  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| SETTRA | 9391# | 9400  | 9401  | 9402  | 9403  | 9405  | 9407  | 9408  | 9409  | 9410  | 9411  | 9412  |       |       |       |
| SETUP  | 1197# | 1941  |       |       |       |       |       |       |       |       |       |       |       |       |       |
| SKIP   | 1197# |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| SLASH  | 1197# |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| SPACE  | 1197# |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| STARS  | 1197# | 1198  | 1202  | 1244  | 1248  | 1441  | 1445  | 1480  | 1500  | 1549  | 1580  | 1584  | 1931  | 1935  | 2089  |
|        | 2093  | 2098  | 2110  | 2123  | 2169  | 2207  | 2275  | 2291  | 2308  | 2325  | 2331  | 2363  | 2382  | 2398  | 2405  |
|        | 2456  | 2528  | 2545  | 2562  | 2568  | 2600  | 2619  | 2635  | 2642  | 2693  | 2766  | 2783  | 2798  | 2805  | 2810  |
|        | 2858  | 2864  | 2885  | 2895  | 2947  | 2962  | 2977  | 2984  | 2989  | 3037  | 3043  | 3064  | 3074  | 3127  | 3138  |
|        | 3151  | 3157  | 3166  | 3258  | 3272  | 3283  | 3296  | 3302  | 3311  | 3403  | 3418  | 3426  | 3442  | 3447  | 3498  |
|        | 3506  | 3522  | 3527  | 3579  | 3597  | 3610  | 3642  | 3647  | 3685  | 3690  | 3783  | 3801  | 3814  | 3846  | 3851  |
|        | 3889  | 3894  | 3988  | 4006  | 4018  | 4034  | 4078  | 4087  | 4093  | 4112  | 4168  | 4190  | 4207  | 4219  | 4235  |
|        | 4279  | 4288  | 4294  | 4313  | 4369  | 4393  | 4406  | 4418  | 4434  | 4478  | 4528  | 4533  | 4573  | 4584  | 4608  |
|        | 4640  | 4695  | 4706  | 4730  | 4762  | 4818  | 4836  | 4860  | 4871  | 4876  | 4943  | 4998  | 5017  | 5041  | 5052  |

|        |       |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
|--------|-------|-------|------|------|-------|-------|------|------|------|------|------|------|------|------|------|
|        | 5057  | 5124  | 5180 | 5195 | 5219  | 5253  | 5272 | 5329 | 5344 | 5368 | 5402 | 5421 | 5479 | 5497 | 5521 |
|        | 5553  | 5560  | 5584 | 5668 | 5686  | 5710  | 5742 | 5749 | 5773 | 5858 | 5872 | 5896 | 5902 | 5909 | 5929 |
|        | 6033  | 6140  | 6151 | 6176 | 6183  | 6235  | 6248 | 6271 | 6280 | 6285 | 6293 | 6305 | 6324 | 6338 | 6358 |
|        | 6408  | 6437  | 6450 | 6473 | 6482  | 6487  | 6495 | 6507 | 6526 | 6540 | 6560 | 6610 | 6640 | 6659 | 6682 |
|        | 6691  | 6720  | 6749 | 6754 | 6809  | 6828  | 6851 | 6860 | 6889 | 6918 | 6923 | 6979 | 6994 | 7018 | 7027 |
|        | 7031  | 7039  | 7045 | 7058 | 7106  | 7116  | 7131 | 7155 | 7164 | 7168 | 7176 | 7182 | 7195 | 7243 | 7255 |
|        | 7271  | 7298  | 7303 | 7309 | 7330  | 7349  | 7404 | 7419 | 7446 | 7451 | 7457 | 7478 | 7497 | 7553 | 7577 |
|        | 7767  | 7791  | 7982 | 7994 | 8096  | 8108  | 8211 | 8215 | 8218 | 8238 | 8262 | 8273 | 8280 | 8285 | 8290 |
|        | 8309  | 8327  | 8383 | 8403 | 8427  | 8438  | 8445 | 8450 | 8455 | 8474 | 8492 | 8550 | 8601 | 8605 | 8656 |
|        | 8660  | 8664  | 8722 | 8763 | 8819  | 8889  | 8966 | 9033 | 9102 | 9117 | 9180 | 9204 | 9272 | 9325 | 9370 |
|        | 9414  | 9418  | 9479 | 9483 | 10117 | 10121 |      |      |      |      |      |      |      |      |      |
| SWITCH | 1063# | 4885  | 5066 | 5591 | 5780  | 5956  | 6062 | 7633 | 7847 |      |      |      |      |      |      |
| SWRSU  | 1197# | 1961# |      |      |       |       |      |      |      |      |      |      |      |      |      |
| TESTAG | 1063# | 2121  | 2319 | 2556 | 2794  | 2973  | 3149 | 3294 | 3437 | 3517 | 3608 | 3812 | 4017 | 4218 | 4417 |
|        | 4595  | 4717  | 4847 | 5028 | 5206  | 5355  | 5508 | 5697 | 5883 | 6162 | 6259 | 6461 | 6670 | 6839 | 7005 |
|        | 7142  | 7282  | 7430 | 7588 | 7802  | 8005  | 8119 | 8249 | 8414 |      |      |      |      |      |      |
| TIMER  | 1063# | 2324  | 2561 | 2797 | 2976  | 3150  | 3295 | 6284 | 6486 | 7030 | 7038 | 7167 | 7175 |      |      |
| TRMTRP | 9391# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| TYPBIN | 1197# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| TYPDEC | 1197# | 8576  | 8583 | 8803 |       |       |      |      |      |      |      |      |      |      |      |
| TYPNAM | 1197# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| TYPNUM | 1197# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| TYPOCS | 1197# | 2005  | 2011 |      |       |       |      |      |      |      |      |      |      |      |      |
| TYPOCT | 1197# | 8775  | 8800 | 9130 |       |       |      |      |      |      |      |      |      |      |      |
| TYPTXT | 1197# | 8572  | 8579 |      |       |       |      |      |      |      |      |      |      |      |      |
| USEREO | 8548# | 8559  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| SSCMRE | 1498# | 1537  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| SSCMTH | 1498# | 1538  | 1539 | 1540 | 1541  | 1542  |      |      |      |      |      |      |      |      |      |
| SSESCA | 1197# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| SSNEWT | 1197# | 2098  | 2291 | 2528 | 2766  | 2947  | 3127 | 3272 | 3418 | 3498 | 3579 | 3783 | 3988 | 4190 | 4393 |
|        | 4573  | 4695  | 4818 | 4998 | 5180  | 5329  | 5479 | 5668 | 5858 | 6140 | 6235 | 6437 | 6640 | 6809 | 6979 |
|        | 7116  | 7255  | 7404 | 7553 | 7767  | 7982  | 8096 | 8218 | 8383 |      |      |      |      |      |      |
| SSSET  | 9391# | 9400  | 9401 | 9402 | 9403  | 9405  | 9407 | 9408 | 9409 | 9410 | 9411 | 9412 |      |      |      |
| SSSKIP | 1197# |       |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .EQUAT | 1063# | 1087  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .HEADE | 1063# | 1064  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SETUP | 1063# | 1931  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SWRHI | 1063# | 1075  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SWRLO | 1063# | 1085# |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SACT1 | 1063# | 1478  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SCATC | 1063# | 1468  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SCMTA | 1063# | 1498  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SEOP  | 1063# | 8548  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SERRO | 1063# | 8720  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SERRT | 1063# | 8761  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SRDOC | 1063# | 9270  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SREAD | 1063# | 9031  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SSAVE | 1063# | 9323  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .SSCOP | 1063# | 8662  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .STRAP | 1063# | 9368  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .STYPO | 1063# | 8964  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .STYPE | 1063# | 8817  |      |      |       |       |      |      |      |      |      |      |      |      |      |
| .STYPO | 1063# | 8887  |      |      |       |       |      |      |      |      |      |      |      |      |      |

CO1

CZRJEB0 DL CTRLR LGC MACY11 30(1046) 04-NOV-77 17:48 PAGE 211  
CZRJEB.P11 04-NOV-77 13:27 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0209

ERRORS DETECTED: 0

RM03:CZRJEB,CZRJEB.SEQ/CRF/SOL/LI:ME/NL:MC:MD:CND=CZRJEB.P11  
RUN-TIME: 36 37 3 SECONDS  
RUN-TIME RATIO: 215/77=2.7  
CORE USED: 31K (61 PAGES)

001