

RP04
RP05, RP06

RP04/5/6 DL CNTLR
CZRJED0

AH-9201D-MC
FICHE 1 OF 2

APR 1982
COPYRIGHT © 76-82
MADE IN USA



The main body of the document is a dense grid of approximately 15 columns and 25 rows of data. Each cell in the grid contains a small, structured table or form. The text within these cells is extremely small and difficult to read, but it appears to be organized into columns and rows, possibly representing a data matrix or a series of related records. The overall appearance is that of a microfiche or a similar high-density data storage format.

RP04
RP05, RP06

RP04/5/6 DL CNTLR
CZRJEDO

AH-9201D-MC
FICHE 2 OF 2

APR 1982
COPYRIGHT © 76-82
MADE IN USA



.REM @

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

IDENTIFICATION

PRODUCT CODE: AC-9200D-MC
PRODUCT NAME: CZRJEDO RP04/5/6 DUAL CONTROLLER TEST, PT 1
PRODUCT DATE: NOVEMBER 1981
MAINTAINER: CX DIAGNOSTIC GROUP
AUTHOR: C. HESS
REVISED BY: MIKE LEAVITT

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976,1977,1982 DIGITAL EQUIPMENT CORPORATION

CONTENTS

1
2
3
4
5
6
7
8
9
10
11
12
13
14

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 PRELIMINARY PROGRAMS
 - 2.3 OTHER PROGRAMS
- 3. LOADING PROCEDURES
- 4. STARTING PROCEDURES

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

- 4.1 STARTING ADDRESSES
- 4.2 UNIBUS & VECTOR ADDRESSES
- 4.3 OPERATOR ACTION
- 5. OPERATING PROCEDURES
 - 5.1 'SOFTWARE' SWITCH REGISTER
 - 5.2 OPERATIONAL SWITCH SETTINGS
 - 5.3 TEST SELECTION
 - 5.4 DUAL PORT TEST CABLE CONNECTION
- 6. ERRORS
- 7. MISCELLANEOUS
 - 7.1 RESTRICTIONS
 - 7.2 LIMITATIONS
 - 7.3 EXECUTION TIME
 - 7.4 STACK POINTER
 - 7.5 SUBROUTINE CALLS
 - 7.6 REQUIRED TESTS
 - 7.7 DISK SURFACE USAGE
 - 7.8 TEST ITERATIONS
 - 7.9 LOOP ON ERROR OPTION
 - 7.10 SPECIAL M7775 'DP' BOARD TESTS
- 8. TEST DESCRIPTIONS
- 9. PROGRAM LISTING

1. ABSTRACT

THE RP04/5/6 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RP04/5/6 DUAL CONTROLLER LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL CONTROLLER MODE IS NOT TESTED BY THIS PROGRAM.

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 PRELIMINARY PROGRAMS

RP04/5/6 DISKLESS CONTROLLER TEST
PART 1 (CZRJG)
PART 2 (CZRJH)

RP04/5/6 FUNCTIONAL CONTROLLER TEST
PART 1 (CZRJI)
PART 2 (CZRJJ)

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

2.3 OTHER PROGRAMS

- A. THE OPERATION OF THE UNLOAD COMMAND AND THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH ARE TESTED BY THE RP04/5/6 DUAL CONTROLLER LOGIC TEST, PART 2 (CZRJF).
- B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RP04/5/6 MULTIDRIVE EXERCISER PROGRAM (CZRJD).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

58 3. LOADING PROCEDURES
59 -----

60 THE PROGRAM MAY BE LOADED BY THE ABSOLUTE PAPER TAPE
61 LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE MEDIA
62 USING THE ASSOCIATED 'XXDP' LOADER. THE PROGRAM MAY NOT
63 BE INCLUDED IN AN 'XXDP' CHAIN.
64

65
66 4. STARTING PROCEDURES
67 -----

68
69 4.1 STARTING ADDRESSES

- 70
71
72 A. THE NORMAL STARTING ADDRESS OF THE PROGRAM IS LOCATION
73 200 (8). STARTING AT THIS ADDRESS ALLOWS THE OPERATOR
74 TO SELECT (OR RESELECT) THE ADDRESS OF THE DRIVE TO BE
75 TESTED.
76
77 B. THE RESTART ADDRESS IS LOCATION 200 (8). THE PROGRAM WILL
78 USE THE CURRENT DRIVE (DCL) ADDRESS.
79
80 C. THE PROGRAM CAN BE STARTED AT LOCATION 204 (8) TO ALLOW THE
81 ADDRESS OF THE RH11 OR RH70 TO BE CHANGED.
82

83 4.2 UNIBUS & VECTOR ADDRESSES

84 THE PROGRAM ASSUMES THE FOLLOWING UNIBUS AND VECTOR ADDRESSES.
85 THESE ADDRESSES MAY BE CHANGED PRIOR TO STARTING THE PROGRAM
86 FROM ANY OF THE STARTING ADDRESSES.
87

88 MEMORY 89 LOCATION	90 CONTENTS	91 FUNCTION
92 -----	93 -----	94 -----
95 1142	177560	TTY KEYBOARD STATUS REG
96 1144	177562	TTY KEYBOARD BUFFER REG
97 1146	177564	TTY PRINTER STATUS REG
98 1150	177566	TTY PRINTER BUFFER REG
99 1210	172540	KW11-P STATUS REG
100 1212	172542	KW11-P COUNTER BUFFER
101 1214	104	KW11-P VECTOR ADDRESS
102 1216	177546	KW11-L STATUS REGISTER
103 1220	100	KW11-L VECTOR ADDRESS

104 4.3 OPERATOR ACTION

- 105 A. CONNECT THE DUAL CONTROLLER TEST CABLE BETWEEN BUS A
106 & BUS B ON THE DRIVE BEING TESTED. (SEE SECTION 5.4)
107 B. LOAD THE PROGRAM INTO MEMORY IN THE PROCESSOR CONTROLLING
108 THE MASSBUS USED FOR TESTING.
109 C. SWITCH THE 'CONTROLLER SELECT' SWITCH ON THE DRIVE TO BE
110 TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
111 D. LOAD THE APPROPRIATE STARTING ADDRESS (200(8), 204(8) OR 210(8)).
112 INTO THE SWITCH REGISTER (OR THE 'SOFTWARE' SWITCH REGISTER,
113 REFER TO SECTION 5.1).
114 E. PRESS START.

- 115 F. ENTER THE DRIVE NUMBER. (THIS MUST BE THE NUMBER DISPLAYED
116 BY THE DRIVE, IF AN RP04, OR THE NUMBER OF THE ADDRESS PLUG IF
117 THE DRIVE IS AN RP05/6).
118 G. ENTER THE NUMBER OF THE TEST TO BE RUN. ('CARRIAGE RETURN'
119 OR '0' WILL RUN ALL TESTS.)
120 H. THE PROGRAM MAY BE STOPPED AT ANY TIME AND RESTARTED
121 FROM LOCATION 200.
122
123

124 5. OPERATING PROCEDURES
125 -----
126
127

128 5.1 'SOFTWARE' SWITCH REGISTER
129

130 IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR
131 THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS
132 NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE
133 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE
134 SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD
135 ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL
136 RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM
137 IS AT A HIGHER PRIORITY PROCESSING AN INTERRUPT. THE
138 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE
139 TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:
140

141 'SWR = NNNNNN NEW ='
142

143 EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER
144 IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED., 'RUBOUT' AND
145 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS
146 DURING SWITCH ENTRY.
147

148 ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH
149 REGISTER MAY BE USED, IF THE PROGRAM FINDS ALL 1'S IN THE SWITCHES.
150 ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER
151 AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.
152

153 5.2 OPERATIONAL SWITCH SETTINGS
154

155 WITH ALL SWITCHES SET TO ZERO, THE PROGRAM WILL TYPE
156 ALL ERRORS AND CONTINUE TESTING.
157

158 THE SWITCH SETTINGS ARE:
159

160 SW<15>=1 HALT ON ERROR
161 SW<14>=1 LOOP ON TEST
162 SW<13>=1 INHIBIT ERROR TYPEOUTS
163 SW<11>=1 INHIBIT TEST ITERATIONS
164 SW<10>=1 RING TTY BELL ON ERROR
165 SW<09>=1 LOOP ON ERROR
166

167 5.3 TEST SELECTION
168

169 INDIVIDUAL TESTS ARE SELECTED IN RESPONSE TO THE 'ENTER
170 TEST NUMBER:' MESSAGE. ANY VALID TEST NUMBER CAN BE
171

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228

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 RP04/5/6 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 RP04 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 M775 FOR RP04'S, OR BY THE ADDRESS PLUG FOR RP05/6'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 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

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

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.

7.5 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

- A. 'SCOPE' (IOT INSTRUCTION). THIS CALL IS PLACED BETWEEN

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

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:

TEST NO.	ITERATION COUNT (IN DECIMAL)
01	1
02	10
03	10
04	1
05	1

343	06	4000
344	07	4000
345	10	100
346	11	100
347	12	4000
348	13	4
349	14	4
350	15	4
351	16	4000
352	17	4000
353	20	4000
354	21	4000
355	22	4000
356	23	4000
357	24	4000
358	25	4000
359	26	4000
360	27	4000
361	30	4000
362	31	4
363	32	4
364	33	4000
365	34	4000
366	35	4
367	36	4
368	37	4
369	40	4
370	41	4000
371	42	4000
372	43	4000
373	44	4000
374	45	20
375	46	20

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 RP04 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR WAS DETECTED.

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.

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
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

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 DESCREPCANCY 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 RP04/5/6, 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.

B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'B'; VERIFY THAT 0'S ARE READ FROM EACH REGISTER.

C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE. MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO NEUTRAL.

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

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

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

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

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

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

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

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

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

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

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

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'

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

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'. 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.

571 8.14 TEST 14 - TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
572
573 VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
574 SEIZING PORT.
575
576 A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
577 SET.
578
579 B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
580
581 C. ISSUE A MASSBUS CLEAR.
582
583 D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION
584 BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
585 'B' IS STILL SET.
586
587 8.15 TEST 15 - TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
588
589 VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
590 SEIZING PORT.
591
592 A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
593 SET.
594
595 B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
596
597 C. ISSUE A MASSBUS CLEAR.
598
599 D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION
600 BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
601 'A' IS STILL SET.
602
603 8.16 TEST 16 - TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
604
605 VERIFY THAT MASSBUS CLEAR DOES NOT RESET ATTENTION BITS WHEN THE
606 DRIVE IS IN NEUTRAL.
607
608 A. SET THE ATTENTION BITS FOR BOTH PORTS.
609
610 B. VERIFY THAT THE DRIVE IS IN NEUTRAL.
611
612 C. ISSUE A MASSBUS INIT. VERIFY THAT NEITHER ATTENTION BIT HAS
613 RESET.
614
615 8.17 TEST 17 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'
616
617 VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
618
619 A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'A'; VERIFY THAT
620 THE DRIVE IS SEIZED.
621
622 B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE
623 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
624
625 8.18 TEST 20 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
626
627 VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.

628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684

- 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 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'

685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741

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'.
- 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

742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798

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.

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

799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855

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.

8.34 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.

856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912

- 8.35 TEST 41 - TEST PORT INTERACTION FROM PORT 'A'
VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 - B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 - D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
 - E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 - F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
 - G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.36 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.
- 8.37 TEST 43 - TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
- A. SET THE ATTENTION BIT FOR PORT 'A'.
 - B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 - C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

913 8.38 TEST 44 - TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
914
915 VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
916
917 A. SET THE ATTENTION BIT FOR PORT 'B'.
918
919 B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
920
921 C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
922 FOR THE DRIVE IS SET.
923
924 8.39 TEST 45 - TEST NO TIMEOUT THROUGH PORT 'A'
925
926 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
927 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
928 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
929 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST
930 BE SELECTED SEPARATELY.
931
932 A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
933
934 B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
935
936 C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE
937 HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
938 SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
939
940 D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
941 BEEN RELEASED.
942
943 E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
944 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
945
946 8.40 TEST 46 - TEST NO TIMEOUT THROUGH PORT 'B'
947
948 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
949 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
950 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
951 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST
952 BE SELECTED SEPARATELY.
953
954 A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
955
956 B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
957
958 C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE
959 HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
960 SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
961
962 D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
963 BEEN RELEASED.
964
965 E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
966 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
967
968
969

970
971
972

9. PROGRAM LISTING
a -----

1

;*LAST REVISION 30-OCT-81

CZ
ER

497
498

```

.TITLE CZRJEDO - DU CNTRL TST 1
:*COPYRIGHT (C) 1976,1977,1981
:*DIGITAL EQUIPMENT CORPORATION
:*COLORADO SPGS., CO. 80919
:*
:*PROGRAM BY C. HESS
:*
:*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
:*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81

```

499

```

.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

```

500
501

.SBTTL BASIC DEFINITIONS

```

001100  :*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
104000  STACK = 1100
000004  ERROR = EMT          ;;BASIC DEFINITION OF ERROR CALL
        SCOPE = IOT       ;;BASIC DEFINITION OF SCOPE CALL

```

```

000011  :*MISCELLANEOUS DEFINITIONS
000012  HT = 11             ;;CODE FOR HORIZONTAL TAB
000015  LF = 12             ;;CODE FOR LINE FEED
000200  CR = 15             ;;CODE FOR CARRIAGE RETURN
177776  CRLF = 200         ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776  PS = 177776       ;;PROCESSOR STATUS WORD
177776  PSW=PS
177774  STKLMT = 177774    ;;STACK LIMIT REGISTER
177772  PIRQ = 177772     ;;PROGRAM INTERRUPT REQUEST REGISTER
177570  DSWR = 177570     ;;HARDWARE SWITCH REGISTER
177570  DDISP = 177570   ;;HARDWARE DISPLAY REGISTER

```

```

000000  :*GENERAL PURPOSE REGISTER DEFINITIONS
000001  R0 = %0             ;;GENERAL REGISTER
000002  R1 = %1             ;;GENERAL REGISTER
000003  R2 = %2             ;;GENERAL REGISTER
000004  R3 = %3             ;;GENERAL REGISTER
000005  R4 = %4             ;;GENERAL REGISTER
000006  R5 = %5             ;;GENERAL REGISTER
000007  R6 = %6             ;;GENERAL REGISTER
000006  R7 = %7             ;;GENERAL REGISTER
000006  SP = %6            ;;STACK POINTER
000007  PC = %7            ;;PROGRAM COUNTER

```

```

000000  :*PRIORITY LEVEL DEFINITIONS
000040  PR0 = 0             ;;PRIORITY LEVEL 0
000100  PR1 = 40            ;;PRIORITY LEVEL 1
000100  PR2 = 100         ;;PRIORITY LEVEL 2

```


BASIC DEFINITIONS

000140	PR3	=	140	::	PRIORITY LEVEL 3
000200	PR4	=	200	::	PRIORITY LEVEL 4
000240	PR5	=	240	::	PRIORITY LEVEL 5
000300	PR6	=	300	::	PRIORITY LEVEL 6
000340	PR7	=	340	::	PRIORITY LEVEL 7

.*'SWITCH REGISTER' SWITCH DEFINITIONS

100000	SW15	=	100000
040000	SW14	=	40000
020000	SW13	=	20000
010000	SW12	=	10000
004000	SW11	=	4000
002000	SW10	=	2000
001000	SW09	=	1000
000400	SW08	=	400
000200	SW07	=	200
000100	SW06	=	100
000040	SW05	=	40
000020	SW04	=	20
000010	SW03	=	10
000004	SW02	=	4
000002	SW01	=	2
000001	SW00	=	1
001000	SW9=SW09		
000400	SW8=SW08		
000200	SW7=SW07		
000100	SW6=SW06		
000040	SW5=SW05		
000020	SW4=SW04		
000010	SW3=SW03		
000004	SW2=SW02		
000002	SW1=SW01		
000001	SW0=SW00		

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

100000	BIT15	=	100000
040000	BIT14	=	40000
020000	BIT13	=	20000
010000	BIT12	=	10000
004000	BIT11	=	4000
002000	BIT10	=	2000
001000	BIT09	=	1000
000400	BIT08	=	400
000200	BIT07	=	200
000100	BIT06	=	100
000040	BIT05	=	40
000020	BIT04	=	20
000010	BIT03	=	10
000004	BIT02	=	4
000002	BIT01	=	2
000001	BIT00	=	1
001000	BIT9=BIT09		
000400	BIT8=BIT08		
000200	BIT7=BIT07		
000100	BIT6=BIT06		
000040	BIT5=BIT05		
000020	BIT4=BIT04		

000010 BIT3=BIT03
 000004 BIT2=BIT02
 000002 BIT1=BIT01
 000001 BIT0=BIT00

```

;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004 ERRVEC = 4           ;; TIME OUT AND OTHER ERRORS
000010 RESVEC = 10        ;; RESERVED AND ILLEGAL INSTRUCTIONS
000014 TBITVEC = 14       ;; "T" BIT
000014 TRTVEC = 14        ;; TRACE TRAP
000014 BPTVEC = 14        ;; BREAKPOINT TRAP (BPT)
000020 IOTVEC = 20        ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024 PWRVEC = 24        ;; POWER FAIL
000030 EMTVEC = 30        ;; EMULATOR TRAP (EMT) **ERROR**
000034 TRAPVEC = 34       ;; "TRAP" TRAP
000060 TKVEC = 60         ;; TTY KEYBOARD VECTOR
000064 TPVEC = 64         ;; TTY PRINTER VECTOR
000240 PIRQVEC = 240      ;; PROGRAM INTERRUPT REQUEST VECTOR
    
```

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

.SBTTL RH11 REGISTERS

;CONTROL AND STATUS REGISTER 1 (RPCS1)

```

000100 IE= 100           ;INTERRUPT ENABLE (BIT #6)
000200 RDY= 200         ;READY (BIT #7)
000400 A16= 400         ;HIGH ORDER BUS ADDRESS BIT (BIT #8)
001000 A17= 1000        ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
002000 PSEL= 2000       ;PORT SELECT (BIT #10)
020000 MCPE= 20000     ;MASSBUS PARITY ERROR (BIT #13)
040000 TRE= 40000      ;TRANSFER ERROR (BIT #14)
100000 SC= 100000      ;SPECIAL CONDITION (BIT #15)
    
```

;WORD COUNT REGISTER (RPWC)
 ;(EACH BIT IS CALLED BY BIT NUMBER)

;BUS ADDRESS REGISTER (RPBA)
 ;(EACH BIT IS CALLED BY BIT NUMBER)

;CONTROL AND STATUS REGISTER 2 (RPCS2)

```

000001 US1= 1           ;UNIT SELECT (BIT #0)
000002 US2= 2           ;UNIT SELECT (BIT #1)
000004 US4= 4           ;UNIT SELECT (BIT #2)
000010 BAI= 10          ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020 PAT= 20          ;MASSBUS PARITY TEST (BIT #4)
000040 CLR= 40          ;CLEAR (BIT #5)
000100 IR= 100          ;INPUT READY (BIT #6)
000200 OR= 200          ;OUTPUT READY (BIT #7)
000400 MPE= 400         ;MASS BUS PARITY ERROR (BIT #8)
001000 MXF= 1000        ;MISSED TRANSFER ERROR (BIT #9)
002000 PGE= 2000       ;PROGRAM ERROR (BIT #10)
004000 NEM= 4000        ;NON EXISTENT MEMORY (BIT #11)
010000 NED= 10000       ;NON EXISTENT DRIVE (BIT #12)
020000 UPE= 20000      ;UNIBUS PARITY ERROR (BIT #13)
040000 WCE= 40000      ;WRITE CHECK ERROR (BIT #14)
100000 DLT= 100000     ;DATA LATE (BIT #15)
    
```

```

541
542      :DATA BUFFER REGISTER (RPDB)
543      :(EACH BIT IS CALLED BY BIT NUMBER)
544
545
546      .SBTTL  RP04/5/6 REGISTERS
547
548      ;CONTROL AND STATUS 1 REGISTER. (#00)
549
550      000001      GO=      1      ;GO BIT (BIT #0)
551      000002      F1=      2      ;FUNCTION CODE BIT #1
552      000004      F2=      4      ;FUNCTION CODE BIT #2
553      000010      F3=     10      ;FUNCTION CODE BIT #3
554      000020      F4=     20      ;FUNCTION CODE BIT #4
555      000040      F5=     40      ;FUNCTION CODE BIT #5
556      004000      DVA=    4000    ;DEVICE AVAILABLE (BIT #11)
557
558      ;DRIVE STATUS REGISTER (RPDS1) (#01)
559
560      ;DF5=      1      DRIVE FORWARD 5"/SEC. (BIT #0)
561      000002      DFF20=  2      ;DRIVE FORWARD 20"/SEC. (BIT #1)
562      000004      DIGB=   4      ;DRIVE TO INNER GUARD BAND (BIT #2)
563      000010      GRV=   10      ;GO REVERSE (BIT #3)
564      000020      DL64=  20      ;DIFFERENCE LESS THAN 64 (BIT #4)
565      000040      DE1=   40      ;DIFFERENCE EQUALS 1 (BIT #5)
566      000100      VV=   100     ;VOLUME VALID (BIT #6)
567      000200      DRY=  200     ;DRIVE READY (BIT #7)
568      000400      DPR=  400     ;DRIVE PRESENT (BIT #8)
569      001000      PGM= 1000     ;PROGRAMABLE (BIT #9)
570      002000      LST= 2000     ;LAST SECTOR TRANSFERRED (BIT #10)
571      004000      WRL= 4000     ;WRITE LOCK (BIT #11)
572      010000      MOL=10000    ;MEDIUM ON-LINE (BIT #12)
573      020000      PIP= 20000   ;POSITIONING OPERATION IN PROGRESS (BIT #13)
574      040000      ERR= 40000   ;COMPOSITE ERROR (BIT #14)
575      100000      ATA=100000   ;ATTENTION ACTIVE (BIT #15)
576
577      ;ERROR REGISTER #01 (RPER1) (#02)
578
579      000001      ILF=      1      ;ILLEGAL FUNCTION (BIT #0)
580      000002      ILR=      2      ;ILLEGAL REGISTER (BIT #1)
581      000004      RMR=      4      ;REGISTER MODIFICATION REFUSED (BIT #2)
582      000010      PAR=     10      ;PARITY ERROR (BIT #3)
583      000020      FER=     20      ;FORMAT ERROR (BIT #4)
584      000040      WCF=     40      ;WRITE CLOCK FAIL (BIT #5)
585      000100      ECH=    100     ;ECC HARD ERROR (BIT #6)
586      000200      HCE=    200     ;HEADER COMPARE ERROR (BIT #7)
587      000400      HCRC=   400     ;HEADER CRC ERROR (BIT #8)
588      001000      AOE=  1000     ;ADDRESS OVERFLOW ERROR (BIT #9)
589      002000      IAE=  2000     ;INVALID ADDRESS ERROR (BIT #10)
590      004000      WLE=  4000     ;WRITE LOCK ERROR (BIT #11)
591      010000      DTE= 10000    ;DRIVE TIMING ERROR (BIT #12)
592      020000      OPI= 20000   ;OPERATION INCOMPLETE (BIT #13)
593      040000      UNS= 40000   ;DRIVE UNSAFE (BIT #14)
594      100000      DCK=100000   ;DATA CHECK ERROR (BIT 15)
595
596      ;MAINTAINABILITY REGISTER (RPMR) (#03)
597

```

598	000001	DMD= 1	:DIAGINOSTIC MODE (BIT #0)
599	000002	MCLK= 2	:MAINTAINABILITY CLOCK (BIT #1)
600	000004	MINX= 4	:MAINTAINABILITY INDEX (BIT #2)
601	000010	MSTCK= 10	:MAINTAINABILITY SECTOR CLOCK (BIT #3)
602	000020	MRD= 20	:MAINTAINABILITY READ (BIT #4)
603	000040	MWR= 40	:MAINTAINABILITY WRITE (BIT #5)
604	000200	DTSY= 200	:MAINTAINABILITY SYNC DETECTED (BIT #7)
605			
606			:ATTENTION SUMMARY PSEUDO-REGISTER (RPAS) (#04)
607			
608	000001	AT0= 1	:DEVICE 0 (BIT #0)
609	000002	AT1= 2	:DEVICE 1 (BIT #1)
610	000004	AT2= 4	:DEVICE 2 (BIT #2)
611	000010	AT3= 10	:DEVICE 3 (BIT #3)
612	000020	AT4= 20	:DEVICE 4 (BIT #4)
613	000040	AT5= 40	:DEVICE 5 (BIT #5)
614	000100	AT6= 100	:DEVICE 6 (BIT #6)
615	000200	AT7= 200	:DEVICE 7 (BIT #7)
616			
617			:DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA) (#05)
618			: (EACH BIT IS CALLED BY BIT NUMBER)
619			
620			:DRIVE TYPE REGISTER (RPDT) (#06)
621			
622	000001	DT00= 1	:DRIVE TYPE NUMBER BIT 1
623	000002	DT01= 2	:DRIVE TYPE NUMBER BIT 2
624	000004	DT02= 4	:DRIVE TYPE NUMBER BIT 3
625	000010	DT03= 10	:DRIVE TYPE NUMBER BIT 4
626	000020	DT04= 20	:DRIVE TYPE NUMBER BIT 5
627	000040	DT05= 40	:DRIVE TYPE NUMBER BIT 6
628	000100	DT06= 100	:DRIVE TYPE NUMBER BIT 7
629	000200	DT07= 200	:DRIVE TYPE NUMBER BIT 8
630	000400	DT08= 400	:DRIVE TYPE NUMBER BIT 9
631	004000	DRQ= 4000	:DRIVE REQUEST REQUIRED (BIT #11)
632	020000	MOH= 20000	:MOVING HEAD (BIT #13)
633	040000	TAP= 40000	:TAPE DRIVE (BIT #14)
634	100000	NBA= 100000	:NOT BLOCK ADDRESSED (BIT #15)
635			
636			:LOOK-AHEAD REGISTER (RPLA) (#07)
637			
638	000001	EXT1= 1	:EXTENSION 1 (BIT #0)
639	000002	EXT2= 2	:EXTENSION 2 (BIT #1)
640	000004	EXT4= 4	:EXTENSION 3 (BIT #2)
641	000010	EXT10= 10	:EXTENSION 4 (BIT #3)
642	000020	EXT20= 20	:EXTENSION 5 (BIT #4)
643	000040	EXT40= 40	:EXTENSION 6 (BIT #5)
644	000100	SC1= 100	:SECTOR COUNT FIELD 0 (BIT #6)
645	000200	SC2= 200	:SECTOR COUNT FIELD 1 (BIT #7)
646	000400	SC4= 400	:SECTOR COUNT FIELD 2 (BIT #8)
647	001000	SC10= 1000	:SECTOR COUNT FIELD 3 (BIT #9)
648	002000	SC20= 2000	:SECTOR COUNT FIELD 4 (BIT #10)
649	004000	TRK1= 4000	:TRACK FIELD 1 (BIT #11)
650	010000	TRK2= 10000	:TRACK FIELD 2 (BIT #12)
651	020000	TRK4= 20000	:TRACK FIELD 3 (BIT #13)
652	040000	TRK10= 40000	:TRACK FIELD 4 (BIT #14)
653	100000	TRK20= 100000	:TRACK FIELD 5 (BIT #15)
654			

```

655      :RP04 ERROR REGISTER #2 (RPER2) (#10)
656
657      000001      WCU=      1      :WRITE CURRENT UNSAFE (BIT #0)
658      000002      CSF=      2      :CURRENT SINK FAILURE (BIT #1)
659      000004      WSU=      4      :WRITE SELECT UNSAFE (BIT #2)
660      000010      CSU=     10      :CURRENT SWITCH UNSAFE (BIT #3)
661      000020      MSE=     20      :MOTOR SEQUENCE ERROR (BIT #4)
662      000040      TDF=     40      :TRANSITIONS DETECTOR FAILURE (BIT #5)
663      000100      TUF=    100      :TRANSITIONS UNSAFE (BIT #6)
664      000200      FEN=    200      :FAILSAFE ENABLED (BIT #7)
665      000400      WRU=    400      :WRITE READY UNSAFE (BIT #8)
666      001000      MHS=   1000      :MULTIPLE HEAD SELECT (BIT #9)
667      002000      NHS=   2000      :NO HEAD SELECTION (BIT #10)
668      004000      IXE=   4000      :INDEX ERROR (BIT #11)
669      010000      VU30=  10000      :30VOLT UNSAFE (BIT #12)
670      020000      PLU=   20000      :PLO UNSAFE (BIT #13)
671      100000      ACU=  100000      :AC UNSAFE (BIT #15)
672
673      :RP05/6 ERROR REGISTER #02 (RPER2) (#10)
674
675      000001      WCU=      1      :WRITE CURRENT UNSAFE (BIT #0)
676      000002      CSF=      2      :CURRENT SINK FAILURE (BIT #1)
677      000004      WSU=      4      :WRITE SELECT UNSAFE (BIT #2)
678      000010      CSU=     10      :CURRENT SWITCH UNSAFE (BIT #3)
679      000020      RAW=     20      :READ AND WRITE (BIT #4)
680      000040      TDF=     40      :TRANSITIONS DETECTOR FAILURE (BIT #5)
681      000100      TUF=    100      :TRANSITIONS UNSAFE (BIT #6)
682      000200      ABS=    200      :ABNORMAL STOP (BIT #7)
683      000400      WRU=    400      :WRITE READY UNSAFE (BIT #8)
684      001000      MHS=   1000      :MULTIPLE HEAD SELECT (BIT #9)
685      002000      NHS=   2000      :NO HEAD SELECTION (BIT #10)
686      004000      IXE=   4000      :INDEX ERROR (BIT #11)
687      020000      PLU=  20000      :PLO UNSAFE (BIT #12)
688
689      :OFFSET REGISTER (RPOF) (#11)
690
691      000001      OF25=     1      :OFFSET 25 MICRO INCHES (BIT #0)
692      000002      OF50=     2      :OFFSET 50 MICRO INCHES (BIT #1)
693      000004      OF100=    4      :OFFSET 100 MICRO INCHES (BIT #2)
694      000010      OF200=   10      :OFFSET 200 MICRO INCHES (BIT #3)
695      000020      OF400=   20      :OFFSET 400 MICRO INCHES (BIT #4)
696      000040      OF800=   40      :OFFSET 800 MICRO INCHES (BIT #5)
697      000200      OFREV=  200      :OFFSET NEGATIVE (REVERSE) (BIT #5)
698      002000      HCI=   2000      :HEADER COMPARE INHIBIT (BIT #10)
699      004000      ECI=   4000      :ERROR CORRECTION CODE INHIBIT (BIT #11)
700      010000      FMT22= 10000      :FORMAT BIT (BIT #12)
701
702      :DESIRED CYLINDER ADDRESS (RPCA) (#12)
703      : (EACH BIT IS CALLED BY BIT NUMBER)
704
705      :CURRENT CYLINDER ADDRESS (RPCC) (#13)
706      : (EACH BIT IS CALLED BY BIT NUMBER)
707
708      :SERIAL NUMBER REGISTER (RPSN) (#14)
709      : (EACH IS CALLED BY BIT NUMBER)
710
711      :RP04 ERROR REGISTER #03 (RPER3) (#15)
  
```

712			
713	000001	PSU= 1	:PACK SPEED UNSAFE (BIT #0)
714	000002	VUF= 2	:VELOCITY UNSAFE (BIT #1)
715	000010	UWR= 10	:ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
716	000020	PRE= 20	:DISK PACK ROTATION ERROR (BIT #4)
717	000040	ACL= 40	:AC LOW (BIT #5)
718	000100	DCL= 100	:DC LOW (BIT #6)
719	040000	SKI= 40000	:SEEK INCOMPLETE (BIT #14)
720	100000	OCYL= 100000	:OFF CYLINDER (BIT #15)
721			
722			:RP05/6 ERROR REGISTER #03 (RPER3) (#15)
723			
724	000001	DCU= 1	:DC UNSAFE (BIT #0)
725	000002	WAO= 2	:WRITE AND OFFSET (BIT #1)
726	000040	ACL= 40	:AC LOW (BIT #5)
727	000100	DCL= 100	:DC LOW (BIT #6)
728	020000	OPE= 20000	:OPERATOR PLUG ERROR (BIT #13)
729	040000	SKI= 40000	:SEEK INCOMPLETE (BIT #14)
730	100000	OCYL= 100000	:OFF CYLINDER ERROR (BIT #15)
731			
732			
733			:ECC POSITION REGISTER (RPEC1) (#16)
734			: (EACH BIT IS CALLED BY BIT NUMBER)
735			
736			
737			:ECC PATTERN REGISTER (RPEC2) (#17)
738			: (EACH BIT IS CALLED BY BIT NUMBER)
739			
740			
741			.SBTTL DEFINITIONS OF THE RH11/RP04/5/6 ADDRESS INDEXES
742			
743	000000	RPCS1=0	:CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
744	000002	RPWC=2	:WORD COUNT REGISTER (NOT A DRIVE REG)
745	000004	RPBA=4	:UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
746	000006	RPDA=6	:DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
747	000010	RPCS2=10	:CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
748	000012	RPDS1=12	:DRIVE STATUS REGISTER (DRIVE REG 01)
749	000014	RPER1=14	:ERROR REGISTER #1 (DRIVE REG. 02)
750	000016	RPAS=16	:ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
751	000020	RPLA=20	:LOOK AHEAD REGISTER (DRIVE REG. 07)
752	000022	RPDB=22	:DATA BUFFER REGISTER (NOT A DRIVE REG.)
753	000024	RPMR=24	:MAINTAINABILITY REGISTER (DRIVE REG. 03)
754	000026	RPDT=26	:DRIVE TYPE REGISTER (DRIVE REG. 06)
755	000030	RPSN=30	:SERIAL NUMBER REGISTER (DRIVE REG. 10)
756	000032	RPOF=32	:OFFSET REGISTER (DRIVE REG. 11)
757	000034	RPCA=34	:DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
758	000036	RPCC=36	:CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
759	000040	RPER2=40	:ERROR REGISTER #2 (DRIVE REG. 14)
760	000042	RPER3=42	:ERROR REGISTER #3 (DRIVE REG. 15)
761	000044	RPEC1=44	:ECC POSITION REGISTER (DRIVE REG. 16)
762	000046	RPEC2=46	:ECC PATTERN REGISTER (DRIVE REG. 17)

```
1          .SBTTL  TRAP CATCHER
          000000
          .=0
          ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
          ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
          ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
          000174 000174
          000174 000000
          000176 000000
          .=174
DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
SWREG:  .WORD 0          ;;SOFTWARE SWITCH REGISTER

          .SBTTL  STARTING ADDRESS(ES)
          000200 000137 002144          JMP    @#START          ;;JUMP TO STARTING ADDRESS OF PROGRAM
2
3 000204 000137 002154          JMP    @#START1         ;;START AND CHANGE THE RH/RP ADDRESS
4
5          .SBTTL  ACT11 HOOKS
          ;*****
          ;HOOKS REQUIRED BY ACT11
          000210          $SVPC=.          ;SAVE PC
          000046          .=46
          056666          $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
          000052          .=52
          020000          .WORD 20000     ;;2)SET LOC.52 TO 20000
          000210          .=$SVPC         ;; RESTORE PC
6
```

0

.SBTTL COMMON TAGS

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

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

.SBTTL USER DEFINED TAGS

001212	172540	\$LKCSR: .WORD	172540	:ADDR OF KW11-P STATUS REGISTER
001214	172542	\$LKCSB: .WORD	172542	:ADDR OF KW11-P COUNTER BUFFER
001216	000104	\$LPVEC: .WORD	104	:ADDR OF KW11-P VECTOR
001220	177546	\$LKS: .WORD	177546	:ADDR OF KW11-L STATUS REGISTER
001222	000100	\$LLVEC: .WORD	100	:ADDR OF KW11-L VECTOR
001224	000000	PORTA: .WORD	0	:ADDRESS OF PORT A
001226	000000	PORTB: .WORD	0	:ADDRESS OF PORT B
001230	000000	PORTC: .WORD	0	:ADDRESS OF DIFFERENT DRIVE
001232	000000	ASR1: .WORD	0	:ATA-A OR ATA-B = 1
001234	000000	PTNBR: .WORD	0	:CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
001236	000000	SEIZPT: .WORD	0	:CONTAINS THE ADDRESS OF THE SEIZING PORT
001240	000000	OPprt: .WORD	0	:CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
001242	000000	TSTNUM: .WORD	0	:NUMBER OF THE CURRENT TEST
001244	000000	CKERR: .WORD	0	:IF -1, A REGISTER MISCOMPARISON OCCURRED
001246	000000	NOSEIZ: .WORD	0	:IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
001250	000000	RELERR: .WORD	0	:IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
001252	000000	TIME: .WORD	0	:ELAPSED TIME COUNTER
001254	000000	WATCH: .WORD	0	:WATCH DOG TIMER LOCATION
001256	000000	TIMEA: .WORD	0	:THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
001260	000000	TIMEAP: .WORD	0	:PORT A TIMEOUT VALUE + 25%
001262	000000	TIMEAM: .WORD	0	:PORT A TIMEOUT VALUE - 25%
001264	000000	TIMEB: .WORD	0	:THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
001266	000000	TIMEBP: .WORD	0	:PORT B TIMEOUT VALUE + 25%
001270	000000	TIMEBM: .WORD	0	:PORT B TIME VALUE - 25%
001272	000000	TIMES: .WORD	0	:STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
001274	000000	KYBCTL: .WORD	0	:SINGLE TEST INDICATOR
001276	000000	CHGADR: .WORD	0	:CHANGE THE RH11 ADDRESS INDICATOR

.SBTTL RH11/RP04/5/6 UNIBUS AND VECTOR ADDRESSES

001300	176700	\$RPADR: .WORD	176700	:RH11/RP04/5/6 UNIBUS ADDRESS
001302	000254	\$RPVEC: .WORD	254	:RH11 INTERRUPT VECTOR ADDRESS

.SBTTL ERROR POINTER TABLE

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

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

1	001304				
2					
3					
4	001304	063446	EM1		:WRONG DRIVE TYPE
5	001306	067504	DH1		
6	001310	071232	DT1		
7	001312	071502	DF1		
8					
9					
10					
11	001314	063467	EM2		:DRIVE NOT ON LINE
12	001316	067504	DH1		
13	001320	071232	DT1		
14	001322	071502	DF1		
15					
16					
17					
18	001324	063511	EM3		:SERIAL NUMBERS NOT THE SAME
19	001326	067555	DH3		
20	001330	071246	DT3		
21	001332	071502	DF1		
22					
23					
24					
25	001334	063573	EM4		:DRIVE NOT SEIZED BY PORT 'N'
26	001336	067624	DH4		
27	001340	071314	DT7		
28	001342	071515	DF7		
29					
30					
31					
32	001344	063624	EM5		:WRONG STATUS SEEN BY THE SEIZING PORT
33	001346	067746	DH5		
34	001350	071262	DT5		
35	001352	071507	DF5		
36					
37					
38					
39	001354	063672	EM6		:REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SE
40	001356	070215	DH13		
41	001360	071334	DT13		
42	001362	071507	DF5		

43				
44			:ERROR 7	
45				
46	001364	063772	EM7	:REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
47	001366	070022	DH7	
48	001370	071314	DT7	
49	001372	071515	DF7	
50				
51			:ERROR 10	
52				
53	001374	064053	EM10	:REGISTER CONTENTS INCORRECT
54	001376	067746	DH5	
55	001400	071262	DT5	
56	001402	071507	DF5	
57				
58			:ERROR 11	
59				
60	001404	064103	EM11	:CONTROL BUS PARITY ERROR WHILE READING REGISTER
61	001406	070144	DH11	
62	001410	071232	DT1	
63	001412	071502	DF1	
64				
65			:ERROR 12	
66				
67	001414	064167	EM12	:DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
68	001416	070704	DH36	
69	001420	071422	DT37	
70	001422	071530	DF36	
71				
72			:ERROR 13	
73				
74	001424	064237	EM13	: 'VOLUME VALID' BIT NOT SET BY READIN PRESET
75	001426	070215	DH13	
76	001430	071334	DT13	
77	001432	071507	DF5	
78				
79			:ERROR 14	
80				
81	001434	064324	EM14	: 'VOLUME VALID' SET ON THE OPPOSITE PORT
82	001436	070215	DH13	
83	001440	071334	DT13	
84	001442	071507	DF5	
85				
86			:ERROR 15	
87				
88	001444	064367	EM15	:THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
89	001446	070022	DH7	
90	001450	071314	DT7	
91	001452	071515	DF7	
92				
93			:ERROR 16	
94				
95	001454	064446	EM16	:ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
96	001456	070022	DH7	
97	001460	071314	DT7	
98	001462	071515	DF7	
99				

100			:ERROR 17	
101				
102	001464	064521	EM17	:ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
103	001466	070022	DH7	
104	001470	071314	DT7	
105	001472	071515	DF7	
106				
107			:ERROR 20	
108				
109	001474	064600	EM20	:DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
110	001476	070704	DH36	
111	001500	071422	DT37	
112	001502	071530	DF36	
113				
114			:ERROR 21	
115				
116	001504	064660	EM21	:DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
117	001506	070704	DH36	
118	001510	071422	DT37	
119	001512	071530	DF36	
120				
121			:ERROR 22	
122				
123	001514	064733	EM22	:DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
124	001516	070334	DH22	
125	001520	071352	DT22	
126	001522	071524	DF31	
127				
128			:ERROR 23	
129				
130	001524	065020	EM23	:TIMEOUT CLEARED THE DRIVE'S ERROR BIT
131	001526	070431	DH23	
132	001530	071364	DT23	
133	001532	071502	DF1	
134				

1			:ERROR 24	
2				
3	001534	065066	EM24	:RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
4	001536	070431	DH23	
5	001540	071364	DT23	
6	001542	071502	DF1	
7				
8				
9			:ERROR 25	
10				
11	001544	065145	EM25	:TIMEOUT ONE-SHOT DID NOT RETRIGGER
12	001546	070704	DH36	
13	001550	071412	DT36	
14	001552	071530	DF36	
15				
16				
17			:ERROR 26	
18				
19	001554	065210	EM26	:DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
20	001556	070334	DH22	
21	001560	071352	DT22	
22	001562	071524	DF31	
23				
24			:ERROR 27	
25				
26	001564	065275	EM27	:REGISTER WRONG AFTER RELEASE WITH REQUEST SET
27	001566	070022	DH7	
28	001570	071314	DT7	
29	001572	071515	DF7	
30				
31			:ERROR 30	
32				
33	001574	065353	EM30	:DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
34	001576	070704	DH36	
35	001600	071412	DT36	
36	001602	071530	DF36	
37				
38			:ERROR 31	
39				
40	001604	065450	EM31	:DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SET
41	001606	070606	DH31	
42	001610	071400	DT31	
43	001612	071524	DF31	
44				
45			:ERROR 32	
46				
47	001614	065525	EM32	:ATTN BIT WRONG AFTER RECALIBRATE COMMAND
48	001616	067746	DH5	
49	001620	071262	DT5	
50	001622	071507	DF5	
51				
52			:ERROR 33	
53				
54	001624	065576	EM33	:DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE S
55	001626	070704	DH36	
56	001630	071412	DT36	
57	001632	071530	DF36	

58				
59			:ERROR 34	
60				
61	001634	065700	EM34	:DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE
62	001636	070704	DH36	
63	001640	071412	DT36	
64	001642	071530	DF36	
65				
66			:ERROR 35	
67				
68	001644	066003	EM35	:DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ONE SH
69	001646	070704	DH36	
70	001650	071422	DT37	
71	001652	071530	DF36	
72				
73			:ERROR 36	
74				
75	001654	066062	EM36	:TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
76	001656	070704	DH36	
77	001660	071412	DT36	
78	001662	071530	DF36	
79				
80			:ERROR 37	
81				
82	001664	066134	EM37	:DRIVE IS NON-EXISTENT
83	001666	070704	DH36	
84	001670	071422	DT37	
85	001672	071530	DF36	
86				
87			:ERROR 40	
88				
89	001674	066202	EM40	:ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
90	001676	067504	DH1	
91	001700	071364	DT23	
92	001702	071502	DF1	
93				
94			:ERROR 41	
95				
96	001704	066257	EM41	:TIMEOUT CLEARED ATTENTION BIT
97	001706	070431	DH23	
98	001710	071364	DT23	
99	001712	071502	DF1	
100				
101			:ERROR 42	
102				
103	001714	066321	EM42	:DRIVE NOT IN NEUTRAL OR SEIZED
104	001716	070733	DH42	
105	001720	071432	DT42	
106	001722	071533	DF42	
107				
108			:ERROR 43	
109				
110	001724	066407	EM43	:DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
111	001726	070733	DH42	
112	001730	071432	DT42	
113	001732	071533	DF42	
114				

115			:ERROR 44	
116				
117	001734	066464	EM44	:WRITE ATTENTION BIT DID NOT SET PORT REQUEST
118	001736	070752	DH44	
119	001740	071400	DT31	
120	001742	071524	DF31	
121				
122			:ERROR 45	
123				
124	001744	066541	EM45	:CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
125	001746	067504	DH1	
126	001750	071232	DT1	
127	001752	071502	DF1	
128				
129			:ERROR 46	
130				
131	001754	066620	FM46	:CAN'T ACCESS DRIVE THROUGH EITHER PORT
132	001756	071047	DH46	
133	001760	071440	DT46	
134	001762	071524	DF31	
135				
136			:ERROR 47	
137				
138	001764	066667	EM47	:ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
139	001766	070431	DH23	
140	001770	071364	DT23	
141	001772	071502	DF1	
142				
143			:ERROR 50	
144				
145	001774	066755	EM50	:ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
146	001776	070215	DH13	
147	002000	071334	DT13	
148	002002	071507	DF5	
149				
150			:ERROR 51	
151				
152	002004	067040	EM51	:ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
153	002006	067746	DH5	
154	002010	071262	DT5	
155	002012	071507	DF5	
156				
157			:ERROR 52	
158				
159	002014	067123	EM52	:ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
160	002016	070215	DH13	
161	002020	071334	DT13	
162	002022	071507	DF5	
163				
164			:ERROR 53	
165				
166	002024	067221	EM53	:CAN'T READ ATTN BIT FROM OPPOSITE PORT
167	002026	070431	DH23	
168	002030	071232	DT1	
169	002032	071502	DF1	
170				
171			:ERROR 54	

172				
173	002034	067302	EM54	;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
174	002036	070334	DH22	
175	002040	071452	DT54	
176	002042	071524	DF31	
177				
178				;ERROR 55
179				
180	002044	067375	EM55	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
181	002046	071144	DH55	
182	002050	071464	DT55	
183	002052	071535	DF55	
184				
185				;ERROR 56
186				
187	002054	067442	EM56	;RH11 DIDN'T RESPOND TO ADDRESSING
188	002056	071222	DH56	
189	002060	071476	DT56	
190	002062	071541	DF56	
191				
193				


```

1      ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
2
3 002064 011600  BADTMO: MOV      (SP),RJ      ;SAVE PC WHERE THE TIME OUT OCCURED
4 002066 005740      TST      -(R0)      ;ADJUST PC -2
5 002070 022626      CMP      (SP)+,(SP)+    ;RESTORE STACK POINTER
6 002072 104401 002100  TYPE     ,65$      ;:TYPE ASCIZ STRING
   002076 000417      BR       64$      ;:GET OVER THE ASCIZ
   ;:65$: .ASCIZ <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
   64$:
7 002136 010046      MOV      R0,-(SP)      ;SETUP FOR TYPING OUT PC
8 002140 104402      TYPOC
9 002142 000240      NOP
   ;:PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
   ;:TO STOP ON UNEXPECTED TIMEOUT.
10
11
12      .SBTTL  START OF PROGRAM
13
14 002144 000240  START:  NOP
15 002146 005037 001276  CLR      CHGADR      ;CLEAR THE 'CHANGE RH/RP ADDRESS' INDICATOR
16 002152 000403      BR       START2     ;GO TO THE START
17
18 002154 012737 177777 001276  START1: MOV     #-1,CHGADR ;SET THE 'CHANGE RH/RP ADDRESS' INDICATOR
19
20 002162 005227 000000  START2: INC     #0      ;TTY LOOP, WAIT FOR INCREMENT
21 002166 001375      BNE     -4          ;OF WORD
22 002170 000005      RESET      ;CLEAR THE WORLD
23
24      .SBTTL  INITIALIZE THE COMMON TAGS
   ;:CLEAR THE COMMON TAGS ($CMTAG) AREA
   MOV     #SCMTAG,R6    ;:FIRST LOCATION TO BE CLEARED
   CLR     (R6)+        ;:CLEAR MEMORY LOCATION
   CMP     #SWR,R6      ;:DONE?
   BNE     -6          ;:LOOP BACK IF NO
   MOV     #STACK,SP    ;:SETUP THE STACK POINTER
   ;:INITIALIZE A FEW VECTORS
   MOV     #SSCOPE,@#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
   MOV     #340,@#IOTVEC+2 ;:LEVEL 7
   MOV     #SEERROR,@#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
   MOV     #340,@#EMTVEC+2 ;:LEVEL 7
   MOV     #STRAP,@#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
   MOV     #340,@#TRAPVEC+2 ;:LEVEL 7
   MOV     $ENDCT,$EOPCT ;:SETUP END-OF-PROGRAM COUNTER
   CLR     $TIMES      ;:INITIALIZE NUMBER OF ITERATIONS
   CLR     $ESCAPE     ;:CLEAR THE ESCAPE ON ERROR ADDRESS
   MOVB   #1,$ERMAX    ;:ALLOW ONE ERROR PER TEST
   MOV     #,$SLPADR   ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
   MOV     #,$SLPERR   ;:SETUP THE ERROR LOOP ADDRESS
   ;:SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
   ;:EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
   MOV     @#ERRVEC,-(SP) ;:SAVE ERROR VECTOR
   MOV     #64$,@#ERRVEC ;:SET UP ERROR VECTOR
   MOV     #DSWR,SWR    ;:SETUP FOR A HARDWARE SWICH REGISTER
   MOV     #DDISP,DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
   CMP     #-1,@SWR    ;:TRY TO REFERENCE HARDWARE SWR
   BNE     66$        ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
   ;:AND THE HARDWARE SWR IS NOT = -1
   BR     65$        ;:BRANCH IF NO TIMEOUT
   64$:  MOV     #65$,(SP) ;:SET UP FOR TRAP RETURN
  
```

```
002362 000002 RTI
002364 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
002372 012737 000174 001142 MOV #DISPREG,DISPLAY
002400 012637 000004 66$: MOV (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR

25 ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
26 002404 012737 002064 000004 MOV #BADTMO,ERRVEC ;;SETUP FOR UNEXPECTED TIMEOUT
27 002412 012737 000300 000006 MOV #PR6,ERRVEC+2 ;;LEVEL 6
28
29 .SBTTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
002420 005227 177777 INC #-1 ;;FIRST TIME?
002424 001037 BNE 67$ ;;BRANCH IF NO
002426 022737 056666 000042 CMP #SENDAD,@#42 ;;ACT-11?
002434 001433 BEQ 67$ ;;BRANCH IF YES
002436 104401 002444 TYPE ,68$ ;;TYPE ASCIZ STRING
002442 000430 BR 67$ ;;GET OVER THE ASCIZ
;;68$: .ASCIZ <CRLF>@CZRJED0 - RP04/5/6 DUAL CONTROLLER TEST, PT 1@<CRLF>
67$:
002524 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
002524 005737 000042 TST @#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
002530 001006 BNE 69$ ;;BRANCH IF YES
002532 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
002540 001005 BNE 70$ ;;BRANCH IF NO
002542 104406 GTSWR ;;GET SOFT-SWR SETTINGS
002544 000403 BR 70$
002546 112737 000001 001134 69$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
002554 70$:
```

```

1          .SBTTL  SELECT PORT ADDRESS
2
3          ;ROUTINE TO CHANGE THE RP/RH BASE ADDRESSES, TO SELECT THE PORT ADDRESS
4          ;NUMBER AND SELECT DYNAMIC DUAL PORT OPTION FOR TEST
5
6 002554 004737 061374 STA: JSR PC,$TKINT ;SETUP THE TTY KEYBOARD
7 002560 004737 003174 JSR PC,CHANGE ;CHECK/CHANGE THE RH/RP ADDRESS
8
9 002564 023737 000042 000046 1$: CMP @#42,@#46 ;ACT11?
10 002572 001414 BEQ 2$ ;BR IF YES
11 002574 104401 063064 TYPE ,ENTERA ;ENTER DRIVE ADDRESS
12 002600 104412 RDOCT ;GET THE ADDRESS
13 002602 012637 001224 MOV (SP)+,PORTA ;STORE THE ADDRESS
14 002606 023727 001224 000007 CMP PORTA,#7 ;SEE IF ADDRESS TOO LARGE
15 002614 101403 BLOS 2$ ;BR IF NOT
16 002616 104401 063113 TYPE ,ADRERR ;TYPE ADDRESS ERROR MESSAGE
17 002622 000760 BR 1$ ;TRY AGAIN
18
19 002624 013737 001224 001226 2$: MOV PORTA,PORTB ;GENERATE THE PORT B ADDRESS
20 002632 005237 001226 INC PORTB ;INCREMENT THE ADDRESS
21 002636 042737 000016 001226 BIC #16,PORTB ;LEAVE BIT 0
22 002644 013746 001224 MOV PORTA,-(SP) ;PUT PORT A ADDRESS ON THE STACK
23 002650 042716 177771 BIC #^C6,(SP) ;SAVE BITS 1 & 2
24 002654 052637 001226 BIS (SP)+,PORTB ;SET BITS 1 & 2 IN PORT B ADDRESS
25 002660 104401 063157 TYPE ,PORTAIS ;'PORT A ADDRESS IS '
26 002664 013746 001224 MOV PORTA,-(SP) ;SAVE PORTA FOR TYPEOUT
    ;TYPE PORT A ADDRESS
    ;GO TYPE--OCTAL ASCII
    002670 104403 TYPOS ;TYPE 1 DIGIT(S)
    002672 001 .BYTE 1 ;SUPPRESS LEADING ZEROS
    002673 000 .BYTE 0 ;'PORT B ADDRESS IS '
27 002674 104401 063206 TYPE ,PORTBIS ;SAVE PORTB FOR TYPEOUT
28 002700 013746 001226 MOV PORTB,-(SP) ;TYPE PORT B ADDRESS
    ;GO TYPE--OCTAL ASCII
    002704 104403 TYPOS ;TYPE 1 DIGIT(S)
    002706 001 .BYTE 1 ;SUPPRESS LEADING ZEROS
    002707 000 .BYTE 0 ;ANOTHER CR-LF
29 002710 104401 001207 TYPE ,$CRLF ;GENERATE ADDRESS OF DRIVE NOT TESTED
30 002714 013737 001224 001230 MOV PORTA,PORTC ;COMPLEMENT SOME BITS
31 002722 062737 000006 001230 ADD #6,PORTC ;SAVE ONLY LOWER BITS
32 002730 042737 177770 001230 BIC #^C7,PORTC ;USE PORT A ADDRESS AS INDEX
33 002736 013701 001224 MOV PORTA,R1 ;GET ATTENTION BIT FOR DRIVE
34 002742 116137 071656 001232 MOVB ATABIT(R1),ASR1 ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
37 002750 005037 001256 CLR TIMEA ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
    002754 005037 001260 CLR TIMEAP ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
    002760 005037 001264 CLR TIMEB ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
    002764 005037 001266 CLR TIMEBP ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
38 002770 004737 056706 JSR PC,CKCLK ;SETUP CLOCK
39 002774 000137 003010 JMP EXEC ;CLOCK HAS BEEN STARTED
40 003000 104401 063235 TYPE ,NOCLOCK ;NO CLOCK ON SYSTEM
41 003004 000000 3$: HALT ;FATAL ERROR
42 003006 000776 BR 3$ ;INTERLOCK THE HALT
43
44          ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
45
46 003010 000005 EXEC: RESET ;CLEAR EVERYTHING
47 003012 005037 177776 CLR PS ;CLEAR THE PROCESSOR STATUS WORD
48 003016 104401 001207 TYPE ,$CRLF ;CR-LF
    
```

49	003022	013700	001300		MOV	\$RPADR,RO		:RH11 ADDRESS FOR INDEXING
50	003026	012706	001100		MOV	#STACK,SP		:LOAD STACK POINTER
51	003032	004737	056706		JSR	PC,CKCLK		:START THE CLOCK
52	003036	000240			NOP			:RETURN IF NO CLOCK
53	003040	004737	061374		JSR	PC,\$TKINT		:INITIALIZE THE KEYBOARD
54	003044	005037	001274		CLR	KYBCTL		:CLEAR SINGLE TEST INDICATOR
55	003050	005037	001100		CLR	\$PASS		:CLEAR THE PASS COUNT
56	003054	112737	000001	001115	MOV	#1,\$ERMAX		:SET ERROR MAX TO 1
57	003062	012737	003062	001106	MOV	#.,\$LPADR		:INITIAL SETTING FOR LOOP ADDRESS
58	003070	012737	003070	001110	MOV	#.,\$LPERR		:INITIAL SETTING FOR LOOP ON ERROR ADDRESS
59	003076	023737	000042	000046	1\$: CMP	@#42,@#46		:ACT11?
60	003104	001405			BEQ	2\$:BR IF YES
61	003106	104401	063302		TYPE	,TESTNO		:ASK FOR TEST NUMBER
62	003112	104412			RDOCT			:GET THE NUMBER
63	003114	012601			MOV	(SP)+,R1		:PUT ENTRY INTO R1
64	003116	001002			BNE	3\$:BR IF NOT ZERO
65	003120	000137	003304		2\$: JMP	TST1AA		:ENTER ZERO - PERFORM ALL TESTS
66								
67	003124	020137	071666		3\$: CMP	R1,MAXTN		:SEE IF NUMBER GREATER THAN MAXIMUM
68	003130	003403			BLE	4\$:BR IF LESS OR EQUAL
69	003132	104401	063322		TYPE	,BADNO		:BAD ENTRY
70	003136	000757			BR	1\$:TRY AGAIN
71	003140	005301			4\$: DEC	R1		:DECREMENT ENTRY
72	003142	006301			ASL	R1		:SHIFT IT LEFT
73	003144	005237	001274		INC	KYBCTL		:SET SINGLE TEST INDICATOR
74	003150	012737	000001	001104	MOV	#1,\$ICNT		:PRESET ITERATION COUNT
75	003156	012746	000240		MOV	#PR5,-(SP)		:PUT NEW PS ON STACK
	003162	012746	003170		MOV	#64\$,-(SP)		:PUT NEW PC ON STACK
	003166	000002			RTI			:POP NEW PC AND PS
	003170				64\$: JMP	@TSTADR(R1)		:JUMP TO TEST
76	003170	000171	071542					
77								
78								:CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
79								
80	003174	005737	001276		CHANGE: TST	CHGADR		:CHANGE THE ADDRESS ?
81	003200	001421			BEQ	3\$:BR IF NOT
82	003202	005037	001276		CLR	CHGADR		:CLEAR THE INDICATOR
83	003206	104401	063351		1\$: TYPE	,ADDRIS		:TYPE OUT WHAT THE PRESENT ADDRESS IS
84	003212	013746	001300		MOV	\$RPADR,-(SP)		:PUT THE ADDRESS ON THE STACK
85	003216	104402			TYPOC			:TYPE THE ACTUAL ADDRESS
86	003220	104401	001207		TYPE	,\$CRLF		:CR-LF
87	003224	104401	063406		TYPE	,NTRH11		:ASK FOR NEW ADDRESS
88	003230	104412			RDOCT			
89	003232	005716			TST	(SP)		:0 OR 'CR' ENTERED ?
90	003234	001402			BEQ	2\$:BR IF EITHER ENTERED (NO ADDRESS CHANGE)
91	003236	011637	001300		MOV	(SP),\$RPADR		:NEW RH11 ADDRESS
92	003242	005726			2\$: TST	(SP)+		:CORRECT THE STACK POINTER
93	003244	012737	003264	000004	3\$: MOV	#4\$,@#4		:LOAD TRAP ADDRESS
94	003252	013700	001300		MOV	\$RPADR,RO		:RH11 ADDRESS
95	003256	005760	000002		TST	RPWC(R0)		:SEE IF RH11 RESPONDS AT THAT ADDRESS
96	003262	000404			BR	5\$:BR, RH11 ALIVE AT PRESENT ADDRESS
97	003264				4\$: EMT	56		
	003264	104056			ADD	#4,SP		:RESET THE STACK POINTER
98	003266	062706	000004		BR	1\$:GET ADDRESS AGAIN
99	003272	000745			MOV	#6,@#4		:RESTORE THE VECTOR
100	003274	012737	000006	000004	5\$: MOV	PC		:RETURN
101	003302	000207			RTS			

TESTS

1
2
15
16
17
18
19

20
21
22
23
31

.SBTTL TESTS

003304 013700 001300
003310 012746 000240
003314 012746 003322
003320 000002
003322

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

64\$:

```
::*****
:*TEST 1 DRIVE ACCESS TEST
:*
:*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
:*
:* A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
:* DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (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.
:*
::*****
```

003322
003322 005737 001274
003326 001406
003330 100002
003332 000137 003010
003336 012737 177777 001274
003344 112737 000001 001102
003352 012737 003374 001106
003360 012737 003374 001110
003366 012737 000001 001176
003374 012706 001100

```
TST1: 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 #1,$TSTNM ;TEST NUMBER
   MOV #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
   MOV #TEST1,$LPERR ;LOAD LOOP ON ERROR ADDRESS
   MOV #1,$TIMES ;DO 1 ITERATION
TEST1: MOV #STACK,SP ;LOAD THE STACK POINTER
```

;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

003400 113760 001224 000010
003406 013737 001224 001234
003414 005760 000012
003420 005037 001244
003424 016037 000010 001126
003432 012737 000010 001122
003440 060037 001122
003444 005037 001124
003450 013737 001126 001164
003456 042737 167777 001164
003464 023737 001124 001164
003472 001414
003474 013737 001126 001174
003502 042737 010000 001174
003510 053737 001174 001124
003516 104037
003520 005137 001244
003524 000240
003526 005737 001244
003532 001403
003534 012760 000040 000010

```
MOV#B PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RPDS1(R0) ;SEE IF DRIVE (PORT A) PRESENT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPCS2(R0),$BDDAT ;GET CONTENTS OF RPCS2
MOV #RPCS2,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$BADDR ;ADD RH11 BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #^CNED,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 64$ ;BR IF OK
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #NED,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
EMT 37
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
64$: NOP
TST CKERR ;WAS 'NED' SET ?
BEQ +10 ;BR IF NOT
MOV #CLR,RPCS2(R0) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
```

CZ
T4

003542	113760	001226	000010	MOV	PORTB,RPCS2(R0)	:SELECT PORT B
003550	013737	001226	001234	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003556	005760	000012		TST	RPDS1(R0)	:SEE IF DRIVE (PORT B) PRESENT
003562	005037	001244		CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
003566	016037	000010	001126	MOV	RPCS2(R0),\$BDDAT	:GET CONTENTS OF RPCS2
003574	012737	000010	001122	MOV	#RPCS2,\$BDADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
003602	060037	001122		ADD	R0,\$BDADR	:ADD RH11 BASE ADDRESS
003606	005037	001124		CLR	\$GDDAT	:WHAT REGISTER SHOULD BE
003612	013737	001126	001164	MOV	\$BDDAT,\$TMP0	:MOVE REGISTER CONTENTS TO '\$TMP0'
003620	042737	167777	001164	BIC	#^CNED,\$TMP0	:SAVE SPECIFIED BITS
003626	023737	001124	001164	CMP	\$GDDAT,\$TMP0	:COMPARE THE BITS
003634	001414			BEQ	66\$:BR IF OK
003636	013737	001126	001174	MOV	\$BDDAT,\$TMP4	:COPY 'BAD DATA'
003644	042737	010000	001174	BIC	#NED,\$TMP4	:CLEAR THE MASKED BITS
003652	053737	001174	001124	BIS	\$TMP4,\$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT
003660	104037			EMT	37	
003662	005137	001244		COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR
003666	000240			NOP		
003670	005737	001244		TST	CKERR	:WAS 'NED' SET ?
003674	001403			BEQ	+10	:BR IF NOT
003676	012760	000040	000010	MOV	#CLR,RPCS2(R0)	:ISSUE MASSBUS INIT TO CLEAR 'NED'

66\$:

:CONFIRM THAT DRIVE IS AN RP04/5/6 AND IS DUAL PORT

32
33
34
38

003704	113760	001224	000010	MOV	PORTA,RPCS2(R0)	:SELECT PORT A
003712	013737	001224	001234	MOV	PORTA,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003720	005037	001244		CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
003724	016037	000026	001126	MOV	RPDT(R0),\$BDDAT	:GET CONTENTS OF RPDT
003732	012737	000026	001122	MOV	#RPDT,\$BDADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
003740	060037	001122		ADD	R0,\$BDADR	:ADD RH11 BASE ADDRESS
003744	012737	024020	001124	MOV	#24020,\$GDDAT	:WHAT REGISTER SHOULD BE
003752	013737	001126	001164	MOV	\$BDDAT,\$TMP0	:MOVE REGISTER CONTENTS TO '\$TMP0'
003760	042737	000003	001164	BIC	#^C177774,\$TMP0	:SAVE SPECIFIED BITS
003766	023737	001124	001164	CMP	\$GDDAT,\$TMP0	:COMPARE THE BITS
003774	001414			BEQ	68\$:BR IF OK
003776	013737	001126	001174	MOV	\$BDDAT,\$TMP4	:COPY 'BAD DATA'
004004	042737	177774	001174	BIC	#177774,\$TMP4	:CLEAR THE MASKED BITS
004012	053737	001174	001124	BIS	\$TMP4,\$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT
004020	104001			EMT	1	
004022	005137	001244		COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR
004026	000240			NOP		
004030	113760	001226	000010	MOV	PORTB,RPCS2(R0)	:SELECT PORT B
004036	013737	001226	001234	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004044	005037	001244		CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
004050	016037	000026	001126	MOV	RPDT(R0),\$BDDAT	:GET CONTENTS OF RPDT
004056	012737	000026	001122	MOV	#RPDT,\$BDADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
004064	060037	001122		ADD	R0,\$BDADR	:ADD RH11 BASE ADDRESS
004070	012737	024020	001124	MOV	#24020,\$GDDAT	:WHAT REGISTER SHOULD BE
004076	013737	001126	001164	MOV	\$BDDAT,\$TMP0	:MOVE REGISTER CONTENTS TO '\$TMP0'
004104	042737	000003	001164	BIC	#^C177774,\$TMP0	:SAVE SPECIFIED BITS
004112	023737	001124	001164	CMP	\$GDDAT,\$TMP0	:COMPARE THE BITS
004120	001414			BEQ	70\$:BR IF OK
004122	013737	001126	001174	MOV	\$BDDAT,\$TMP4	:COPY 'BAD DATA'
004130	042737	177774	001174	BIC	#177774,\$TMP4	:CLEAR THE MASKED BITS
004136	053737	001174	001124	BIS	\$TMP4,\$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT
004144	104001			EMT	1	
004146	005137	001244		COM	CKERR	:SET THE REGISTER COMPARE ERROR INDICATOR

68\$:

39
40
41
46

```

004152 000240          70$:  NOP
                                ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
004154 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT A
004162 013737 001224 001234  MOV   PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004170 005037 001244          CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
004174 016037 000012 001126  MOV   RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
004202 012737 000012 001122  MOV   #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004210 060037 001122          ADD   R0, $BDADR    ;ADD RH11 BASE ADDRESS
004214 012737 001000 001124  MOV   #PGM, $GDDAT  ;WHAT REGISTER SHOULD BE
004222 013737 001126 001164  MOV   $BDDAT, $STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
004230 042737 176777 001164  BIC   #^CPGM, $STMP0 ;SAVE SPECIFIED BITS
004236 023737 001124 001164  CMP   $GDDAT, $STMP0 ;COMPARE THE BITS
004244 001414          BEQ   72$          ;BR IF OK
004246 013737 001126 001174  MOV   $BDDAT, $STMP4 ;COPY 'BAD DATA'
004254 042737 001000 001174  BIC   #PGM, $STMP4  ;CLEAR THE MASKED BITS
004262 053737 001174 001124  BIS   $STMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004270 104045          EMT   45
004272 005137 001244          COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
004276 000240          72$:  NOP
                                CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
004300 005037 001244          MOV   RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
004304 016037 000012 001126  MOV   #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004312 012737 000012 001122  ADD   R0, $BCADR    ;ADD RH11 BASE ADDRESS
004320 060037 001122          MOV   #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
004324 012737 010600 001124  MOV   $BDDAT, $STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
004332 013737 001126 001164  BIC   #^C10600, $STMP0 ;SAVE SPECIFIED BITS
004340 042737 167177 001164  CMP   $GDDAT, $STMP0 ;COMPARE THE BITS
004346 023737 001124 001164  BEQ   74$          ;BR IF OK
004354 001414          MOV   $BDDAT, $STMP4 ;COPY 'BAD DATA'
004356 013737 001126 001174  BIC   #10600, $STMP4 ;CLEAR THE MASKED BITS
004364 042737 010600 001174  BIS   $STMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004372 053737 001174 001124  EMT   2
004400 104002          COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
004402 005137 001244          74$:  NOP
                                MOVB  PORTB,RPCS2(R0) ;SELECT PORT B
004410 113760 001226 000010  MOV   PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004416 013737 001226 001234  CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
004424 005037 001244          MOV   RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
004430 016037 000012 001126  MOV   #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004436 012737 000012 001122  ADD   R0, $BDADR    ;ADD RH11 BASE ADDRESS
004444 060037 001122          MOV   #PGM, $GDDAT  ;WHAT REGISTER SHOULD BE
004450 012737 001000 001124  MOV   $BDDAT, $STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
004456 013737 001126 001164  BIC   #^CPGM, $STMP0 ;SAVE SPECIFIED BITS
004464 042737 176777 001164  CMP   $GDDAT, $STMP0 ;COMPARE THE BITS
004472 023737 001124 001164  BEQ   76$          ;BR IF OK
004500 001414          MOV   $BDDAT, $STMP4 ;COPY 'BAD DATA'
004502 013737 001126 001174  BIC   #PGM, $STMP4  ;CLEAR THE MASKED BITS
004510 042737 001000 001174  BIS   $STMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004516 053737 001174 001124  EMT   45
004524 104045          COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
004526 005137 001244          76$:  NOP
                                CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
004532 000240          MOV   RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
004534 005037 001244          MOV   #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004540 016037 000012 001126  ADD   R0, $BDADR    ;ADD RH11 BASE ADDRESS
004546 012737 000012 001122
004554 060037 001122

```

```

004560 012737 010600 001124      MOV      #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
004566 013737 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
004574 042737 167177 001164      BIC      #^C10600,$TMP0    ;SAVE SPECIFIED BITS
004602 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
004610 001414                      BEQ      78$               ;BR IF OK
004612 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
004620 042737 010600 001174      BIC      #10600,$TMP4     ;CLEAR THE MASKED BITS
004626 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
004634 104002                      EMT      2
004636 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
004642 000240      78$:      NOP

47
48      ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
49
50 004644 113760 001224 000010      MOVVB   PORTA,RPCS2(R0)   ;SELECT PORT A
51 004652 016037 000030 001124      MOV     RPSN(R0),$GDDAT   ;STORE THE PORT A SERIAL NUMBER
52 004660 113760 001226 000010      MOVVB   PORTB,RPCS2(R0)   ;SELECT PORT B
53 004666 016037 000030 001126      MOV     RPSN(R0),$BDDAT   ;STORE THE PORT B SERIAL NUMBER
54 004674 023737 001124 001126      CMP     $GDDAT,$BDDAT    ;ARE THEY THE SAME ?
55 004702 001406                      BEQ     1$                ;BR IF THEY ARE
56 004704 104003                      EMT     3
57 004706 032777 100000 174224      BIT     #SW15,@SWR        ;HALT ON ERROR ?
58 004714 001001                      BNE     1$                ;BR IF SET - PROGRAM HAS ALREADY HALTED
59 004716 000000                      HALT
60 004720 000004      1$:      SCOPE             ;LOOP ?
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80

```

```

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

```

```

004722
004722 005737 001274      TST2:   TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
004726 001406                      BEQ     2$                ;BR IF NOT
004730 100002                      BPL     1$                ;BR IF JUST ENTERED TEST
004732 000137 003010      JMP     EXEC              ;RETURN & GET NEXT TEST NUMBER
004736 012737 177777 001274      1$:     MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
004744 112737 000002 001102      2$:     MOVVB   #2,$STNM      ;TEST NUMBER
004752 012737 004774 001106      MOV     #TEST2,$LPADR    ;LOAD LOOP ON TEST ADDRESS
004760 012737 004774 001110      MOV     #TEST?, $LPERR   ;LOAD LOOP ON ERROR ADDRESS
004766 012737 000012 001176      MOV     #10,$TIMES      ;DO 10. ITERATIONS
81 004774 012706 001100      TEST2:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
147 005000 012737 000240 177776      MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING

```



```

005006 005037 001256 CLR TIMEA ;CLEAR TIMEOUT VALUE FOR PORT A
005012 005037 001260 CLR TIMEAP ;CLEAR UPPER TIMEOUT TOLERANCE
005016 005037 001262 CLR TIMEAM ;CLEAR LOWER TIMEOUT TOLERANCE
  
```

;START THE TIMER

```

005022 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
005026 012737 003720 001254 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

005034 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
005042 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
005050 005060 000012 CLR RPDS1(R0) ;WRITE RPDS1
005054 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
005062 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005070 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
005076 016037 000012 001126 MOV RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
005104 010037 001122 MOV RO,$BDADR ;RH11 BASE ADDRESS
005110 062737 000012 001122 ADD #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
005116 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
005122 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
005130 001403 BEQ 64$ ;BR IF IT IS
005132 104004 EMT 4
005134 000137 006320 JMP 5$ ;BYPASS REST OF THE SUBTEST
  
```

64\$:

```

005140 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
005146 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005154 016037 000012 001126 MOV RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
005162 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
005170 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
005176 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
005202 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
005210 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
005216 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
005224 001401 BEQ 65$ ;BR IF THEY ARE
005226 104005 EMT 5
005230 000240 NOP
  
```

65\$:

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

```

005232 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
005240 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005246 016046 000046 MOV RPEC2(R0),-(SP) ;STORE REGISTER RPEC2, PORT B, FOR CHECK
005252 016046 000044 MOV RPEC1(R0),-(SP) ;STORE REGISTER RPEC1, PORT B, FOR CHECK
005256 016046 000042 MOV RPER3(R0),-(SP) ;STORE REGISTER RPER3, PORT B, FOR CHECK
005262 016046 000030 MOV RPSN(R0),-(SP) ;STORE REGISTER RPSN, PORT B, FOR CHECK
005266 016046 000036 MOV RPCC(R0),-(SP) ;STORE REGISTER RPCC, PORT B, FOR CHECK
005272 016046 000034 MOV RPCA(R0),-(SP) ;STORE REGISTER RPCA, PORT B, FOR CHECK
005276 016046 000032 MOV RPOF(R0),-(SP) ;STORE REGISTER RPOF, PORT B, FOR CHECK
005302 016046 000040 MOV RPER2(R0),-(SP) ;STORE REGISTER RPER2, PORT B, FOR CHECK
005306 016046 000020 MOV RPLA(R0),-(SP) ;STORE REGISTER RPLA, PORT B, FOR CHECK
005312 016046 000026 MOV RPDT(R0),-(SP) ;STORE REGISTER RPDT, PORT B, FOR CHECK
005316 016046 000006 MOV RPDA(R0),-(SP) ;STORE REGISTER RPDA, PORT B, FOR CHECK
005322 016046 000024 MOV RPMR(R0),-(SP) ;STORE REGISTER RPMR, PORT B, FOR CHECK
005326 016046 000014 MOV RPER1(R0),-(SP) ;STORE REGISTER RPER1, PORT B, FOR CHECK
  
```

```

;WAIT FOR PCRT A TO TIMEOUT

005332 005760 000012 1$: TST RPDS1(R0) ;WAIT FOR THE DRIVE TO TIMEOUT
005336 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
005340 005737 001254 TST WATCH ;CHECK WATCH
005344 001372 BNE 1$ ;BR IF NOT ZERO
005346 104036 EMT 36
005350 000137 005740 JMP 4$ ;BYPASS TIMEOUT TIME CHECK
005354 012737 000340 177776 2$: MOV #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
005362 013737 001252 001256 MOV TIME,TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
005370 004537 057072 JSR R5,TOLER ;CALCULATE THE TOLERANCE
005374 001256 .WORD TIMEA ;TIMEOUT VALUE FOR PORT A
005376 012637 001260 MOV (SP)+,TIMEAP ;+25% TOLERANCE
005402 012637 001262 MOV (SP)+,TIMEAM ;-25% TOLERANCE

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

005406 023727 001252 000764 CMP TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
005414 103001 BHIS 3$ ;BR IF IT WAS
005416 104055 EMT 55

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

005420 012737 000240 177776 3$: MOV #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

;VERIFY THAT THE DRIVE IS IN NEUTRAL

005426 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
005432 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
005440 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
005444 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
005452 113760 001224 000010 MOV#B PORTA,RPCS2(R0) ;SELECT PORT A.
005460 016037 000012 001170 MOV RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
005466 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
005474 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005502 113760 001226 000010 MOV#B PORTB,RPCS2(R0) ;SELECT PORT B.
005510 016037 000012 001172 MOV RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
005516 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
005524 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005532 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
005540 001006 BNE 66$ ;BR IF NOT
005542 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
005546 001037 BNE 68$ ;BR IF NOT
005550 104046 EMT 46
005552 000137 005736 JMP 70$ ;BYPASS THE REST OF THE CHECKS
005556 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
005564 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005572 113760 001226 000010 MOV#B PORTB,RPCS2(R0) ;SELECT PORT B.
005600 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
005604 001414 BEQ 67$ ;BR IF ZERO
005606 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005614 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
005622 113760 001224 000010 MOV#B PORTA,RPCS2(R0) ;SELECT PORT A.
005630 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
005634 001004 BNE 68$ ;BR IF NOT
005636 012737 177777 001250 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
    
```

```

005644 104022      EMT      22
005646 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
005654 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
005662 042737 100100 001170      BIC      #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
005670 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
005676 001401      BEQ      69$ ;BR IF OK FROM PORT A.
005700 104007      EMT      7
005702 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
005710 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
005716 042737 100100 001172      BIC      #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
005724 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
005732 001401      BEQ      70$ ;BR IF OK
005734 104007      EMT      7
005736 000240      70$:  NOP
    
```

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

```

005740 013737 001226 001234 4$:  MOV      PORTB,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
005746 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER1
005752 062737 000014 001122      ADD      #RPER1,$BDADR ;ADDRESS OF RPER1 FOR TYPEOUT
005760 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
005764 001401      BEQ      .+4 ;CONTENTS ZERO ?
005766 104006      EMT      6
005770 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPMR
005774 062737 000024 001122      ADD      #RPMR,$BDADR ;ADDRESS OF RPMR FOR TYPEOUT
006002 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPMR
006006 001401      BEQ      .+4 ;CONTENTS ZERO ?
006010 104006      EMT      6
006012 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPDA
006016 062737 000006 001122      ADD      #RPDA,$BDADR ;ADDRESS OF RPDA FOR TYPEOUT
006024 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDA
006030 001401      BEQ      .+4 ;CONTENTS ZERO ?
006032 104006      EMT      6
006034 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPDT
006040 062737 000026 001122      ADD      #RPDT,$BDADR ;ADDRESS OF RPDT FOR TYPEOUT
006046 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDT
006052 001401      BEQ      .+4 ;CONTENTS ZERO ?
006054 104006      EMT      6
006056 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPLA
006062 062737 000020 001122      ADD      #RPLA,$BDADR ;ADDRESS OF RPLA FOR TYPEOUT
006070 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPLA
006074 001401      BEQ      .+4 ;CONTENTS ZERO ?
006076 104006      EMT      5
006100 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER2
006104 062737 000040 001122      ADD      #RPER2,$BDADR ;ADDRESS OF RPER2 FOR TYPEOUT
006112 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER2
006116 001401      BEQ      .+4 ;CONTENTS ZERO ?
006120 104006      EMT      6
006122 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPOF
006126 062737 000032 001122      ADD      #RPOF,$BDADR ;ADDRESS OF RPOF FOR TYPEOUT
006134 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPOF
006140 001401      BEQ      .+4 ;CONTENTS ZERO ?
006142 104006      EMT      6
006144 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPCA
006150 062737 000034 001122      ADD      #RPCA,$BDADR ;ADDRESS OF RPCA FOR TYPEOUT
006156 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPCA
    
```

```

006162 001401 BEQ .+4 ;CONTENTS ZERO ?
006164 104006 EMT 6
006166 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPCC
006172 062737 000036 001122 ADD #RPCC,$BDADR ;ADDRESS OF RPCC FOR TYPEOUT
006200 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPCC
006204 001401 BEQ .+4 ;CONTENTS ZERO ?
006206 104006 EMT 6
006210 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPSN
006214 062737 000030 001122 ADD #RPSN,$BDADR ;ADDRESS OF RPSN FOR TYPEOUT
006222 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPSN
006226 001401 BEQ .+4 ;CONTENTS ZERO ?
006230 104006 EMT 6
006232 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER3
006236 062737 000042 001122 ADD #RPER3,$BDADR ;ADDRESS OF RPER3 FOR TYPEOUT
006244 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER3
006250 001401 BEQ .+4 ;CONTENTS ZERO ?
006252 104006 EMT 6
006254 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPEC1
006260 062737 000044 001122 ADD #RPEC1,$BDADR ;ADDRESS OF RPEC1 FOR TYPEOUT
006266 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC1
006272 001401 BEQ .+4 ;CONTENTS ZERO ?
006274 104006 EMT 6
006276 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPEC2
006302 062737 000046 001122 ADD #RPEC2,$BDADR ;ADDRESS OF RPEC2 FOR TYPEOUT
006310 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC2
006314 001401 BEQ .+4 ;CONTENTS ZERO ?
006316 104006 EMT 6
006320 000004 5$: SCOPE ;LOOP ?
  
```

165
166

```

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

```

006322 005737 001274 TST3: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
006322 001406 BEQ 2$ ;BR IF NOT
006330 100002 BPL 1$ ;BR IF JUST ENTERED TEST
006332 000137 003010 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
006336 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
006344 112737 000003 001102 2$: MOVB #3,$TSTNM ;TEST NUMBER
006352 012737 006374 001106 MOV #TEST3,$LPADR ;LOAD LOOP ON TEST ADDRESS
006360 012737 006374 001110 MOV #TEST3,$LPERR ;LOAD LOOP ON ERROR ADDRESS
006366 012737 000012 001176 MOV #10,$TIMES ;DO 10. ITERATIONS
  
```

```

167 006374 012706 001100 TEST3: MOV #STACK,SP ;LOAD THE STACK POINTER
168 006400 012737 000240 177776 MOV #<5*32.>,@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
006406 005037 001264 CLR TIMEB ;CLEAR TIMEOUT VALUE FOR PORT B
006412 005037 001266 CLR TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
006416 005037 001270 CLR TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

006422 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
006426 012737 003720 001254 MOV #2000.,WATCH ;SET WATCH TO 2000 MS

;SEIZE THE DRIVE THROUGH PORT B

006434 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
006442 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
006450 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
006454 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
006462 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006470 013737 001224 001240 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
006476 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
006504 010037 001122 MOV RO,$BDADR ;RH11 BASE ADDRESS
006510 062737 000012 001122 ADD #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
006516 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
006522 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
006530 001403 BEQ 64$ ;BR IF IT IS
006532 104004 EMT 4
006534 000137 007720 JMP 5$ ;BYPASS REST OF THE SUBTEST
006540 64$:
006540 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
006546 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006554 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
006562 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
006570 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
006576 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
006602 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
006610 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
006616 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
006624 001401 BEQ 65$ ;BR IF THEY ARE
006626 104005 EMT 5
006630 65$: NOP

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

006632 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
006640 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006646 016046 000046 MOV RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
006652 016046 000044 MOV RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
006656 016046 000042 MOV RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
006662 016046 000030 MOV RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT A, FOR CHECK
006666 016046 000036 MOV RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT A, FOR CHECK
006672 016046 000034 MOV RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT A, FOR CHECK
006676 016046 000032 MOV RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT A, FOR CHECK
006702 016046 000040 MOV RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
006706 016046 000020 MOV RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT A, FOR CHECK
006712 016046 000026 MOV RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT A, FOR CHECK
006716 016046 000006 MOV RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT A, FOR CHECK
  
```

```

006722 016046 000024      MOV      RPMR(RO),-(SP)  ;STORE REGISTER RPMR, PORT A, FOR CHECK
006726 016046 000014      MOV      RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK

                                ;WAIT FOR PORT B TO TIMEOUT

006732 005760 000012      1$:     TST      RPDS1(RO)  ;WAIT FOR THE DRIVE TO TIMEOUT
006736 001006              BNE      2$             ;BR WHEN TIMEOUT OCCURS
006740 005737 001254      TST      WATCH         ;CHECK WATCH
006744 001372              BNE      1$             ;BR IF NOT ZERO
006746 104036              EMT      36
006750 000137 007340      JMP      4$             ;BYPASS TIMEOUT TIME CHECK
006754 012737 000340 177776 2$:     MOV      #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
006762 013737 001252 001264      MOV      TIME,TIMEB    ;SAVE THE ELAPSED TIME FOR PORT B
006770 004537 057072      JSR      R5,TOLER      ;CALCULATE THE TOLERANCE
006774 001264              .WORD    TIMEB         ;TIMEOUT VALUE FOR PORT B
006776 012637 001266      MOV      (SP)+,TIMEBP  ;+25% TOLERANCE
007002 012637 001270      MOV      (SP)+,TIMEBM  ;-25% TOLERANCE

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

007006 023727 001252 000764      CMP      TIME,#500.    ;WAS MEASURED TIME AT LEAST 500 MS?
007014 103001              BHS      3$             ;BR IF IT WAS
007016 104055              EMT      55

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

00702C 012737 000240 177776 3$:     MOV      #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

007026 005037 001250      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
007032 012737 000012 001122      MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
007040 060037 001122      ADD      RO,$BDADR     ;ADD THE I/O BASE ADDRESS
007044 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
007052 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
007060 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
007066 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
007074 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007102 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
007110 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
007116 013737 001172 001166      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
007124 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007132 023737 001164 001166      CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
007140 001006              BNE      66$           ;BR IF NOT
007142 005737 001164      TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
007146 001037              BNE      68$           ;BR IF NOT
007150 104046              EMT      46
007152 000137 007336      JMP      70$           ;BYPASS THE REST OF THE CHECKS
007156 013737 001170 001126 66$:     MOV      $TMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
007164 013737 001226 001234      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007172 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
007200 005737 001164      TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
007204 001414              BEQ      67$           ;BR IF ZERO
007206 013737 001224 001234      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007214 013737 001172 001126      MOV      $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
007222 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
007230 005737 001164      TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
    
```

```

007234 001004      BNE      68$      ;BR IF NOT
007236 012737 177777 001250 67$:  MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
007244 104022      EMT      22
007246 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
007254 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
007262 042737 100100 001170      BIC      #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
007270 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
007276 001401      BEQ      69$      ;BR IF OK FROM PORT A.
007300 104007      EMT      7
007302 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
007310 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
007316 042737 100100 001172      BIC      #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
007324 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
007332 001401      BEQ      70$      ;BR IF OK
007334 104007      EMT      7
007336 000240      NOP
    
```

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

```

007340 013737 001224 001234 4$:  MOV      PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007346 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER1
007352 062737 000014 001122      ADD      #RPER1,$BDADR ;ADDRESS OF RPER1 FOR TYPEOUT
007360 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
007364 001401      BEQ      .+4 ;CONTENTS ZERO ?
007366 104006      EMT      6
007370 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPMR
007374 062737 000024 001122      ADD      #RPMR,$BDADR ;ADDRESS OF RPMR FOR TYPEOUT
007402 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPMR
007406 001401      BEQ      .+4 ;CONTENTS ZERO ?
007410 104006      EMT      6
007412 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPDA
007416 062737 000006 001122      ADD      #RPDA,$BDADR ;ADDRESS OF RPDA FOR TYPEOUT
007424 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDA
007430 001401      BEQ      .+4 ;CONTENTS ZERO ?
007432 104006      EMT      6
007434 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPDT
007440 062737 000026 001122      ADD      #RPDT,$BDADR ;ADDRESS OF RPDT FOR TYPEOUT
007446 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDT
007452 001401      BEQ      .+4 ;CONTENTS ZERO ?
007454 104006      EMT      6
007456 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPLA
007462 062737 000020 001122      ADD      #RPLA,$BDADR ;ADDRESS OF RPLA FOR TYPEOUT
007470 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPLA
007474 001401      BEQ      .+4 ;CONTENTS ZERO ?
007476 104006      EMT      6
007500 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER2
007504 062737 000040 001122      ADD      #RPER2,$BDADR ;ADDRESS OF RPER2 FOR TYPEOUT
007512 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER2
007516 001401      BEQ      .+4 ;CONTENTS ZERO ?
007520 104006      EMT      6
007522 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPOF
007526 062737 000032 001122      ADD      #RPOF,$BDADR ;ADDRESS OF RPOF FOR TYPEOUT
007534 012637 001126      MOV      (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPOF
007540 001401      BEQ      .+4 ;CONTENTS ZERO ?
007542 104006      EMT      6
007544 010037 001122      MOV      R0,$BDADR ;BASE ADDRESS FOR REGISTER RPCA
    
```

```

007550 062737 000034 001122 ADD #RPCA,$BDADR ;ADDRESS OF RPCA FOR TYPEOUT
007556 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPCA
007562 001401 BEQ .+4 ;CONTENTS ZERO ?
007564 104006 EMT 6
007566 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPCC
007572 062737 000036 001122 ADD #RPCC,$BDADR ;ADDRESS OF RPCC FOR TYPEOUT
007600 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPCC
007604 001401 BEQ .+4 ;CONTENTS ZERO ?
007606 104006 EMT 6
007610 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPSN
007614 062737 000030 001122 ADD #RPSN,$BDADR ;ADDRESS OF RPSN FOR TYPEOUT
007622 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPSN
007626 001401 BEQ .+4 ;CONTENTS ZERO ?
007630 104006 EMT 6
007632 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPER3
007636 062737 000042 001122 ADD #RPER3,$BDADR ;ADDRESS OF RPER3 FOR TYPEOUT
007644 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER3
007650 001401 BEQ .+4 ;CONTENTS ZERO ?
007652 104006 EMT 6
007654 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPEC1
007660 062737 000044 001122 ADD #RPEC1,$BDADR ;ADDRESS OF RPEC1 FOR TYPEOUT
007666 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC1
007672 001401 BEQ .+4 ;CONTENTS ZERO ?
007674 104006 EMT 6
007676 010037 001122 MOV R0,$BDADR ;BASE ADDRESS FOR REGISTER RPEC2
007702 062737 000046 001122 ADD #RPEC2,$BDADR ;ADDRESS OF RPEC2 FOR TYPEOUT
007710 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPEC2
007714 001401 BEQ .+4 ;CONTENTS ZERO ?
007716 104006 EMT 6
007720 000004 5$: SCOPE ;LOOP ?
    
```

169
187
188

```

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

```

007722 005737 001274 TST4: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
007726 001406 BEQ 2$ ;BR IF NOT
007730 100002 BPL 1$ ;BR IF JUST ENTERED TEST
007732 000137 003010 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
007736 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
007744 112737 000004 001102 2$: MOVB #4,$TSTNM ;TEST NUMBER
    
```



```

007752 012737 007774 001106      MOV      #TEST4,$LPADR      ;LOAD LOOP ON TEST ADDRESS
007760 012737 007774 001110      MOV      #TEST4,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
007766 012737 000001 001176      MOV      #1,$TIMES        ;DO 1 ITERATION
189 007774 012706 001100      TEST4:  MOV      #STACK,SP   ;LOAD THE STACK POINTER
233 010000 113760 001224 000010      MOV      PORTA,RPCS2(R0)   ;SELECT PORT A
010006 013737 001224 001234      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;START THE TIMER

010014 005037 001252      CLR      TIME              ;CLEAR THE ELAPSED TIME COUNTER
010020 012737 003720 001254      MOV      #2000.,WATCH     ;SET WATCH TO 2000 MS
010026 013737 001224 001236      MOV      PORTA,SEIZPT     ;'SEIZED' PORT ADDRESS

;ISSUE DRIVE CLEAR COMMAND

010034 012760 000011 000000      MOV      #11,RPCS1(R0)    ;ISSUE A DRIVE CLEAR

;VERIFY THAT DRIVE SEIZED BY PORT A.

010042 113760 001226 000010      MOV      PORTB,RPCS2(R0)  ;SELECT PORT B
010050 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010056 005037 001244      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
010062 016037 000012 001126      MOV      RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
010070 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010076 060037 001122      ADD      R0,$BDADR       ;ADD RH11 BASE ADDRESS
010102 005037 001124      CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
010106 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
010114 001403      BEQ     64$              ;BR IF OK
010116 104012      EMT     12
010120 005137 001244      COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
010124 000240      64$:  NOP
010126 113760 001224 000010      MOV      PORTA,RPCS2(R0)  ;SELECT PORT A
010134 013737 001224 001234      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010142 005037 001244      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
010146 016037 000012 001126      MOV      RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
010154 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010162 060037 001122      ADD      R0,$BDADR       ;ADD RH11 BASE ADDRESS
010166 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
010174 013737 001126 001164      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010202 042737 106177 001164      BIC     #^C71600,$TMP0   ;SAVE SPECIFIED BITS
010210 023737 001124 001164      CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010216 001414      BEQ     66$              ;BR IF OK
010220 013737 001126 001174      MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
010226 042737 071600 001174      BIC     #71600,$TMP4    ;CLEAR THE MASKED BITS
010234 053737 001174 001124      BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
010242 104010      EMT     10
010244 005137 001244      COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
010250 000240      66$:  NOP
010252 005037 001244      CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
010256 016037 000000 001126      MOV     RPCS1(R0),$BDDAT ;GET CONTENTS OF RPCS1
010264 012737 000000 001122      MOV     #RPCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010272 060037 001122      ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
010276 012737 004210 001124      MOV     #4210,$GDDAT    ;WHAT REGISTER SHOULD BE
010304 013737 001126 001164      MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010312 042737 100000 001164      BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
010320 023737 001124 001164      CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010326 001414      BEQ     68$              ;BR IF OK
  
```

```

010330 013737 001126 001174      MOV      $BDDAT,$STMP4      ;COPY 'BAD DATA'
010336 042737 077777 001174      BIC      #77777,$STMP4      ;CLEAR THE MASKED BITS
010344 053737 001174 001124      BIS      $STMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
010352 104010                               EMT      10
010354 005137 001244                               COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
010360 000240                               68$:    NOP
    
```

;ISSUE READIN PRESET COMMAND AND SET FMT22

```

234 010362 012760 000023 000000      MOV      #23,RPCS1(R0)      ;ISSUE A READIN PRESET
010370 012760 010000 000032      MOV      #FMT22,RPOF(R0)    ;SET FMT22
    
```

;VERIFY THAT THE DRIVE STATUS IS CORRECT

```

010376 005037 001244                               CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
010402 016037 000012 001126      MOV      RPDS1(R0),$BDDAT    ;GET CONTENTS OF RPDS1
010410 012737 000012 001122      MOV      #RPDS1,$BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010416 060037 001122                               ADD      R0,$BDADR          ;ADD RH11 BASE ADDRESS
010422 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
010430 013737 001126 001164      MOV      $BDDAT,$STMP0      ;MOVE REGISTER CONTENTS TO '$STMP0'
010436 042737 106077 001164      BIC      #^C71700,$STMP0    ;SAVE SPECIFIED BITS
010444 023737 001124 001164      CMP      $GDDAT,$STMP0      ;COMPARE THE BITS
010452 001414                               BEQ      70$                ;BR IF OK
010454 013737 001126 001174      MOV      $BDDAT,$STMP4      ;COPY 'BAD DATA'
010462 042737 071700 001174      BIC      #71700,$STMP4      ;CLEAR THE MASKED BITS
010470 053737 001174 001124      BIS      $STMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
010476 104013                               EMT      13
010500 005137 001244                               COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
010504 000240                               70$:    NOP
010506 113760 001226 000010      MOV      PORTB,RPCS2(R0)    ;SELECT PORT B
010514 013737 001226 001234      MOV      PORTB,PTNBR        ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

;WAIT FOR TIMEOUT TO RELEASE DRIVE

```

010522 005760 000012                               1$:    TST      RPDS1(R0)        ;WAIT FOR THE PORT TO TIME OUT
010526 001006                               BNE      2$                ;BR WHEN TIMEOUT OCCURS
010530 005737 001254                               TST      WATCH              ;CHECK THE WATCH
010534 001372                               BNE      1$                ;BR IF NOT ZERO
010536 104036                               EMT      36
010540 000137 011056                               JMP      3$                ;BYPASS ATTN REGISTER CHECK
    
```

;SEE IF DRIVE RETURNED TO NEUTRAL

```

010544                               2$:
    
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

010544 005037 001250                               CLR      RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
010550 012737 000012 001122      MOV      #RPDS1,$BDADR      ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
010556 060037 001122                               ADD      R0,$BDADR          ;ADD THE I/O BASE ADDRESS
010562 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
010570 113760 001224 000010      MOV      PORTA,RPCS2(R0)    ;SELECT PORT A.
010576 016037 000012 001170      MOV      RPDS1(R0),$STMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
010604 013737 001170 001164      MOV      $STMP2,$STMP0      ;COPY IT INTO '$STMP0'
010612 042737 100100 001164      BIC      #ATA!VV,$STMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010620 113760 001226 000010      MOV      PORTB,RPCS2(R0)    ;SELECT PORT B.
010626 016037 000012 001172      MOV      RPDS1(R0),$STMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
    
```

```

010634 013737 001172 001166      MOV      $TMP3,$TMP1      :COPY IT INTO '$TMP1'
010642 042737 100100 001166      BIC      #ATA!VV,$TMP1    :CLEAR PORT DEPENDENT BITS FROM THE COPY
010650 023737 001164 001166      CMP      $TMP0,$TMP1     :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
010656 001006                BNE      72$             :BR IF NOT
010660 005737 001164                TST      $TMP0           :REGISTERS ARE THE SAME: ARE THEY ZERO ?
010664 001037                BNE      74$             :BR IF NOT
010666 104046                EMT      46              :
010670 000137 011054                JMP      76$             :BYPASS THE REST OF THE CHECKS
010674 013737 001170 001126 72$:    MOV      $TMP2,$BDDAT     :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
010702 013737 001226 001234        MOV      PORTB,PTNBR     :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010710 113760 001226 000010        MOVB     PORTB,RPCS2(R0) :SELECT PORT B.
010716 005737 001164                TST      $TMP0           :SEE IF STATUS EQ 0 FROM PORT A.
010722 001414                BEQ      73$             :BR IF ZERO
010724 013737 001224 001234        MOV      PORTA,PTNBR     :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010732 013737 001172 001126        MOV      $TMP3,$BDDAT     :'BAD DATA' FOR ERROR TYPE OUT
010740 113760 001224 000010        MOVB     PORTA,RPCS2(R0) :SELECT PORT A.
010746 005737 001166                TST      $TMP1           :SEE IF STATUS EQ ZERO FROM PORT B.
010752 001004                BNE      74$             :BR IF NOT
010754 012737 177777 001250 73$:    MOV      #-1,RELERR      :SET 'RELEASE ERROR' INDICATOR
010762 104022                EMT      22              :
010764 013737 001170 001126 74$:    MOV      $TMP2,$BDDAT     :LOOK FOR BIT FAILURES WHEN RPDS1 READ
010772 013737 001224 001234        MOV      PORTA,PTNBR     :CHANGE PORT NUMBER
011000 042737 100100 001170        BIC      #ATA!VV,$TMP2    :DON'T CHECK ATTN BIT OR VV BIT
011006 023737 001124 001170        CMP      $GDDAT,$TMP2    :ALL BITS OK ?
011014 001401                BEQ      75$             :BR IF OK FROM PORT A.
011016 104007                EMT      7               :
011020 013737 001172 001126 75$:    MOV      $TMP3,$BDDAT     :CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
011026 013737 001226 001234        MOV      PORTB,PTNBR     :CHANGE PORT NUMBER
011034 042737 100100 001172        BIC      #ATA!VV,$TMP3    :DON'T CHECK ATTN BIT OR VV BIT
011042 023737 001124 001172        CMP      $GDDAT,$TMP3    :SEE IF READ OK FROM PORT B.
011050 001401                BEQ      76$             :BR IF OK
011052 104007                EMT      7               :
011054 000240                NOP                      :
011056 000004                3$:      SCOPE           :LOOP ?
    
```

250
251

```

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

```

011060                TST5:
011060 005737 001274                TST      KYBCTL          :PERFORMING ONLY SINGLE TESTS ?
011064 001406                BEQ      2$             :BR IF NOT
011066 100002                BPL      1$             :BR IF JUST ENTERED TEST
011070 000137 003010                JMP      EXEC           :RETURN & GET NEXT TEST NUMBER
011074 012737 177777 001274 1$:    MOV      #-1,KYBCTL     :SET SINGLE TEST INDICATOR
    
```

```

011102 112737 000005 001102 2$:   MOVB   #5,$STSTNM   ;TEST NUMBER
011110 012737 011132 001106       MOV   #TEST5,$LPADR ;LOAD LOOP ON TEST ADDRESS
011116 012737 011132 001110       MOV   #TEST5,$LPERR ;LOAD LOOP ON ERROR ADDRESS
011124 012737 000001 001176       MOV   #1,$TIMES     ;DO 1 ITERATION
252 011132 012706 001100 TEST5: MOV   #STACK,SP    ;LOAD THE STACK POINTER
253 011136 113760 001226 000010     MOVB  PORTB,$RPCS2(R0) ;SELECT PORT B
011144 013737 001226 001234       MOV   PORTB,$PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;START THE TIMER

011152 005037 001252           CLR   TIME          ;CLEAR THE ELAPSED TIME COUNTER
011156 012737 003720 001254       MOV   #2000,$WATCH ;SET WATCH TO 2000 MS
011164 013737 001226 001236       MOV   PORTB,$SEIZPT ;'SEIZED' PORT ADDRESS

;ISSUE DRIVE CLEAR COMMAND

011172 012760 000011 000000       MOV   #11,$RPCS1(R0) ;ISSUE A DRIVE CLEAR

;VERIFY THAT DRIVE SEIZED BY PORT B.

011200 113760 001224 000010       MOVB  PORTA,$RPCS2(R0) ;SELECT PORT A
011206 013737 001224 001234       MOV   PORTA,$PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011214 005037 001244           CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
011220 016037 000012 001126       MOV   RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
011226 012737 000012 001122       MOV   #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011234 060037 001122           ADD   R0,$BDADR     ;ADD RH11 BASE ADDRESS
011240 005037 001124           CLR   $GDDAT        ;WHAT REGISTER SHOULD BE
011244 023737 001124 001126       CMP   $GDDAT,$BDDAT ;IS THE REGISTER OK ?
011252 001403           BEQ   64$          ;BR IF OK
011254 104012           EMT   12
011256 005137 001244           COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
011262 000240 64$:  NOP
011264 113760 001226 000010       MOVB  PORTB,$RPCS2(R0) ;SELECT PORT B
011272 013737 001226 001234       MOV   PORTB,$PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011300 005037 001244           CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
011304 016037 000012 001126       MOV   RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
011312 012737 000012 001122       MOV   #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011320 060037 001122           ADD   R0,$BDADR     ;ADD RH11 BASE ADDRESS
011324 012737 011600 001124       MOV   #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
011332 013737 001126 001164       MOV   $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
011340 042737 106177 001164       BIC   #^C71600,$STMP0 ;SAVE SPECIFIED BITS
011346 023737 001124 001164       CMP   $GDDAT,$STMP0 ;COMPARE THE BITS
011354 001414           BEQ   66$          ;BR IF OK
011356 013737 001126 001174       MOV   $BDDAT,$STMP4 ;COPY 'BAD DATA'
011364 042737 071600 001174       BIC   #71600,$STMP4 ;CLEAR THE MASKED BITS
011372 053737 001174 001124       BIS   $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
011400 104010           EMT   10
011402 005137 001244           COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
011406 000240 66$:  NOP
011410 005037 001244           CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
011414 016037 000000 001126       MOV   RPCS1(R0),$BDDAT ;GET CONTENTS OF RPCS1
011422 012737 000000 001122       MOV   #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011430 060037 001122           ADD   R0,$BDADR     ;ADD RH11 BASE ADDRESS
011434 012737 004210 001124       MOV   #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
011442 013737 001126 001164       MOV   $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
011450 042737 100000 001164       BIC   #^C77777,$STMP0 ;SAVE SPECIFIED BITS
011456 023737 001124 001164       CMP   $GDDAT,$STMP0 ;COMPARE THE BITS
    
```

```

011464 001414          BEQ      68$          :BR IF OK
011466 013737 001126 001174  MOV     $BDDAT,$TMP4  :COPY 'BAD DATA'
011474 042737 077777 001174  BIC     #77777,$TMP4  :CLEAR THE MASKED BITS
011502 053737 001174 001124  BIS     $TMP4,$GDDAT  : 'OR' WITH GOOD DATA FOR TYPEOUT
011510 104010          EMT     10
011512 005137 001244          COM     CKERR        :SET THE REGISTER COMPARE ERROR INDICATOR
011516 000240          NOP
  
```

68\$:

;ISSUE READIN PRESET COMMAND AND SET FMT22

```

011520 012760 000023 000000  MOV     #23,RPCS1(RO) :ISSUE A READIN PRESET
011526 012760 010000 000032  MOV     #FMT22,RPOF(RO) :SET FMT22
  
```

257

;VERIFY THAT THE DRIVE STATUS IS CORRECT

```

011534 005037 001244          CLR     CKERR         :CLEAR THE 'CHECK ERROR' INDICATOR
011540 016037 000012 001126  MOV     RPDS1(RO),$BDDAT :GET CONTENTS OF RPDS1
011546 012737 000012 001122  MOV     #RPDS1,$BDADR  :FORM REGISTER ADDRESS OF ERROR MESSAGE
011554 060037 001122          ADD     RO,$BDADR     :ADD RH11 BASE ADDRESS
011560 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT :WHAT REGISTER SHOULD BE
011566 013737 001126 001164  MOV     $BDDAT,$TMP0  :MOVE REGISTER CONTENTS TO '$TMP0'
011574 042737 106077 001164  BIC     #^C71700,$TMP0 :SAVE SPECIFIED BITS
011602 023737 001124 001164  CMP     $GDDAT,$TMP0  :COMPARE THE BITS
011610 001414          BEQ     70$          :BR IF OK
011612 013737 001126 001174  MOV     $BDDAT,$TMP4  :COPY 'BAD DATA'
011620 042737 071700 001174  BIC     #71700,$TMP4  :CLEAR THE MASKED BITS
011626 053737 001174 001124  BIS     $TMP4,$GDDAT  : 'OR' WITH GOOD DATA FOR TYPEOUT
011634 104013          EMT     13
011636 005137 001244          COM     CKERR        :SET THE REGISTER COMPARE ERROR INDICATOR
011642 000240          NOP
011644 113760 001224 000010  MOV     PORTA,RPCS2(RO) :SELECT PORT A
011652 013737 001224 001234  MOV     PORTA,PTNBR   :MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

70\$:

;WAIT FOR TIMEOUT TO RELEASE DRIVE

```

011660 005760 000012          1$:   TST     RPDS1(RO)   :WAIT FOR THE PORT TO TIME OUT
011664 001006          BNE     2$          :BR WHEN TIMEOUT OCCURS
011666 005737 001254          TST     WATCH      :CHECK THE WATCH
011672 001372          BNE     1$          :BR IF NOT ZERO
011674 104036          EMT     36
011676 000137 012214          JMP     3$          :BYPASS ATTN REGISTER CHECK
  
```

;SEE IF DRIVE RETURNED TO NEUTRAL

011702

2\$:

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

011702 005037 001250          CLR     RELERR       :CLEAR THE 'RELEASE ERROR' INDICATOR
011706 012737 000012 001122  MOV     #RPDS1,$BDADR  :FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
011714 060037 001122          ADD     RO,$BDADR     :ADD THE I/O BASE ADDRESS
011720 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT :COMPARISON CONSTANT
011726 113760 001224 000010  MOV     PORTA,RPCS2(RO) :SELECT PORT A.
011734 016037 000012 001170  MOV     RPDS1(RO),$TMP2 :GET THE DRIVE STATUS REGISTER FROM PORT A.
011742 013737 001170 001164  MOV     $TMP2,$TMP0  :COPY IT INTO '$TMP0'
011750 042737 100100 001164  BIC     #ATA!VV,$TMP0  :CLEAR PORT DEPENDENT BITS FROM THE COPY
011756 113760 001226 000010  MOV     PORTB,RPCS2(RO) :SELECT PORT B.
  
```

```

011764 016037 000012 001172      MOV      RPDS1(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
011772 013737 001172 001166      MOV      $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
012000 042737 100100 001166      BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012006 023737 001164 001166      CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
012014 001006                                BNE      72$            ;BR IF NOT
012016 005737 001164                                TST      $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
012022 001037                                BNE      74$            ;BR IF NOT
012024 104046                                EMT      46
012026 000137 012212                                JMP      76$            ;BYPASS THE REST OF THE CHECKS
012032 013737 001170 001126 72$:  MOV      $TMP2, $BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012040 013737 001226 001234      MOV      PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012046 113760 001226 000010      MOV      PORTB, RPCS2(R0) ;SELECT PORT B.
012054 005737 001164                                TST      $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
012060 001414                                BEQ      73$            ;BR IF ZERO
012062 013737 001224 001234      MOV      PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012070 013737 001172 001126      MOV      $TMP3, $BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
012076 113760 001224 000010      MOV      PORTA, RPCS2(R0) ;SELECT PORT A.
012104 005737 001166                                TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
012110 001004                                BNE      74$            ;BR IF NOT
012112 012737 177777 001250 73$:  MOV      #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
012120 104026                                EMT      26
012122 013737 001170 001126 74$:  MOV      $TMP2, $BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
012130 013737 001224 001234      MOV      PORTA, PTNBR   ;CHANGE PORT NUMBER
012136 042737 100000 001170      BIC      #ATA, $TMP2    ;DON'T CHECK THE ATTN BIT
012144 023737 001124 001170      CMP      $GDDAT, $TMP2  ;ALL BITS OK ?
012152 001401                                BEQ      75$            ;BR IF OK FROM PORT A.
012154 104007                                EMT      7
012156 013737 001172 001126 75$:  MOV      $TMP3, $BDDAT  ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
012164 013737 001226 001234      MOV      PORTB, PTNBR   ;CHANGE PORT NUMBER
012172 042737 100000 001172      BIC      #ATA, $TMP3    ;DON'T CHECK THE ATTN BIT
012200 023737 001124 001172      CMP      $GDDAT, $TMP3  ;SEE IF READ OK FROM PORT B.
012206 001401                                BEQ      76$            ;BR IF OK
012210 104007                                EMT      7
012212 000240                                NOP
012214 000004 3$:      SCOPE ;LOOP ?
    
```

258
270
271

```

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

```

012216                                TST6:
012216 005737 001274      TST      KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
012222 001406                                BEQ      2$             ;BR IF NOT
012224 100002                                BPL      1$             ;BR IF JUST ENTERED TEST
012226 000137 003010      JMP      EXEC           ;RETURN & GET NEXT TEST NUMBER
012232 012737 177777 001274 1$:  MOV      #-1, KYBCTL    ;SET SINGLE TEST INDICATOR
012240 112737 000006 001102 2$:  MOV      #6, $TSTNM     ;TEST NUMBER
012246 012737 012270 001106      MOV      #TEST6, $LPADR ;LOAD LOOP ON TEST ADDRESS
    
```

```

012254 012737 012270 001110      MOV    #TEST6,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
012262 012737 007640 001176      MOV    #4000,$TIMES    ;DO 4000. ITERATIONS
272 012270 012706 001100      TEST6: MOV    #STACK,SP  ;LOAD THE STACK POINTER
299
;START THE TIMER

012274 005037 001252      CLR    TIME            ;CLEAR THE ELAPSED TIME COUNTER
012300 012737 003720 001254      MOV    #2000.,WATCH   ;SET WATCH TO 2000 MS

;SEIZE THE DRIVE THROUGH PORT A

012306 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
012314 013737 001224 001236      MOV    PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
012322 005060 000012      CLR    RPDS1(R0)     ;WRITE RPDS1
012326 013737 001226 001240      MOV    PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS

;RELEASE THE DRIVE FROM PORT A

012334 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
012342 013737 001224 001234      MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012350 012760 000013 000000      MOV    #13,RPDS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

012356 005037 001250      CLR    RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
012362 012737 000012 001122      MOV    #RPDS1,$BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
012370 060037 001122      ADD    R0,$BDADR     ;ADD THE I/O BASE ADDRESS
012374 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
012402 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
012410 016037 000012 001170      MOV    RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012416 013737 001170 001164      MOV    $TMP2,$TMP0   ;COPY IT INTO 'TMP0'
012424 042737 100100 001164      BIC    #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012432 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
012440 016037 000012 001172      MOV    RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
012446 013737 001172 001166      MOV    $TMP3,$TMP1   ;COPY IT INTO 'TMP1'
012454 042737 100100 001166      BIC    #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012462 023737 001164 001166      CMP    $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
012470 001006      BNE    66$          ;BR IF NOT
012472 005737 001164      TST    $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
012476 001037      BNE    68$          ;BR IF NOT
012500 104046      EMT    46
012502 000137 012666      JMP    70$          ;BYPASS THE REST OF THE CHECKS
012506 013737 001170 001126 66$: MOV    $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012514 013737 001226 001234      MOV    PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012522 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
012530 005737 001164      TST    $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
012534 001414      BEQ    67$          ;BR IF ZERO
012536 013737 001224 001234      MOV    PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012544 013737 001172 001126      MOV    $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
012552 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
012560 005737 001166      TST    $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
012564 001004      BNE    68$          ;BR IF NOT
012566 012737 177777 001250 67$: MOV    #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
012574 104026      EMT    26
012576 013737 001170 001126 68$: MOV    $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ

```

```

012604 013737 001224 001234      MOV    PORTA,PTNBR      ;CHANGE PORT NUMBER
012612 042737 100000 001170      BIC    #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
012620 023737 001124 001170      CMP    $GDDAT,$TMP2    ;ALL BITS OK ?
012626 001401                      BEQ    69$              ;BR IF OK FROM PORT A.
012630 104007                      EMT    7
012632 013737 001172 001126 69$:  MOV    $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
012640 013737 001226 001234      MOV    PORTB,PTNBR     ;CHANGE PORT NUMBER
012646 042737 100000 001172      BIC    #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
012654 023737 001124 001172      CMP    $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
012662 001401                      BEQ    70$              ;BR IF OK
012664 104007                      EMT    7
012666 000240                      NOP
012670 005737 001250 70$:      TST    RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
012674 001402                      BEQ    .+6              ;BR IF IN NEUTRAL
012676 000137 013152                      JMP    1$              ;GO WAIT FOR DRIVE TO TIMEOUT
012702 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
012710 013737 001224 001234      MOV    PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012716 005037 001244                      CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
012722 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
012730 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
012736 060037 001122                      ADD    R0,$BDADR       ;ADD RH11 BASE ADDRESS
012742 005037 001124                      CLR    $GDDAT          ;WHAT REGISTER SHOULD BE
012746 013737 001126 001164      MOV    $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
012754 042737 077777 001164      BIC    #^CATA,$TMP0    ;SAVE SPECIFIED BITS
012762 023737 001124 001164      CMP    $GDDAT,$TMP0    ;COMPARE THE BITS
012770 001414                      BEQ    71$              ;BR IF OK
012772 013737 001126 001174      MOV    $BDDAT,$TMP4    ;COPY 'BAD DATA'
013000 042737 100000 001174      BIC    #ATA,$TMP4      ;CLEAR THE MASKED BITS
013006 053737 001174 001124      BIS    $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
013014 104017                      EMT    17
013016 005137 001244 71$:      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
013022 000240                      NOP
013024 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
013032 013737 001226 001234      MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013040 005037 001244                      CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
013044 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
013052 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
013060 060037 001122                      ADD    R0,$BDADR       ;ADD RH11 BASE ADDRESS
013064 005037 001124                      CLR    $GDDAT          ;WHAT REGISTER SHOULD BE
013070 013737 001126 001164      MOV    $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
013076 042737 077777 001164      BIC    #^CATA,$TMP0    ;SAVE SPECIFIED BITS
013104 023737 001124 001164      CMP    $GDDAT,$TMP0    ;COMPARE THE BITS
013112 001414                      BEQ    73$              ;BR IF OK
013114 013737 001126 001174      MOV    $BDDAT,$TMP4    ;COPY 'BAD DATA'
013122 042737 100000 001174      BIC    #ATA,$TMP4      ;CLEAR THE MASKED BITS
013130 053737 001174 001124      BIS    $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
013136 104017                      EMT    17
013140 005137 001244 73$:      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
013144 000240                      NOP
013146 000137 013204                      JMP    2$              ;GO CHECK FOR LOOP ON ERROR

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

013152 113760 001226 000010 1$:      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
013152 013737 001226 001234      MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013160

```



```

013166 005760 000012      TST  RPDS1(R0)      ;WAIT FOR TIMEOUT TO RELEASE DRIVE
013172 001004              BNE  2$             ;BR WHEN DRIVE RELEASED
013174 005737 001254      TST  WATCH          ;CHECK THE WATCH
013200 001364              BNE  1$             ;BR IF NOT ZERO
013202 104036              EMT  36
013204 000004      2$:  SCOPE                ;LOOP ?

```

311
312

```

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

```

```

013206 005737 001274      TST7:  TST  KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
013206 001406              BEQ  2$             ;BR IF NOT
013212 100002              BPL  1$             ;BR IF JUST ENTERED TEST
013216 000137 003010      JMP  EXEC          ;RETURN & GET NEXT TEST NUMBER
013222 012737 177777 001274 1$:  MOV  #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
013230 112737 000007 001102 2$:  MOVB #7,$STNM      ;TEST NUMBER
013236 012737 013260 001106      MOV  #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
013244 012737 013260 001110      MOV  #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
013252 012737 007640 001176      MOV  #4000,$TIMES  ;DO 4000. ITERATIONS
313 013260 012706 001100      TEST7: MOV  #STACK,SP ;LOAD THE STACK POINTER
314

```

313
314

:START THE TIMER

```

013264 005037 001252      CLR  TIME          ;CLEAR THE ELAPSED TIME COUNTER
013270 012737 003720 001254      MOV  #2000.,WATCH ;SET WATCH TO 2000 MS

```

:SEIZE THE DRIVE THROUGH PORT B

```

013276 113760 001226 000010      MOVB  PORTB,RPCS2(R0) ;SELECT PORT B
013304 013737 001226 001236      MOV  PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
013312 005060 000012              CLR  RPDS1(R0)      ;WRITE RPDS1
013316 013737 001224 001240      MOV  PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS

```

:RELEASE THE DRIVE FROM PORT B

```

013324 113760 001226 000010      MOVB  PORTB,RPCS2(R0) ;SELECT PORT B
013332 013737 001226 001234      MOV  PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013340 012760 000013 000000      MOV  #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B

```

:VERIFY THAT THE DRIVE IS IN NEUTRAL

```

013346 005037 001250              CLR  RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
013352 012737 000012 001122      MOV  #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
013360 060037 001122              ADD  R0,$BDADR     ;ADD THE I/O BASE ADDRESS
013364 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT

```

013372	113760	001224	000010		MOV B	PORTA,RPCS2(R0)	:SELECT PORT A.
013400	016037	000012	001170		MOV	RPDS1(R0),\$TMP2	:GET THE DRIVE STATUS REGISTER FROM PORT A.
013406	013737	001170	001164		MOV	\$TMP2,\$TMP0	:COPY IT INTO '\$TMP0'
013414	042737	100100	001164		BIC	#ATA!VV,\$TMP0	:CLEAR PORT DEPENDENT BITS FROM THE COPY
013422	113760	001226	000010		MOV B	PORTB,RPCS2(R0)	:SELECT PORT B.
013430	016037	000012	001172		MOV	RPDS1(R0),\$TMP3	:GET THE DRIVE STATUS REGISTER FROM PORT B.
013436	013737	001172	001166		MOV	\$TMP3,\$TMP1	:COPY IT INTO '\$TMP1'
013444	042737	100100	001166		BIC	#ATA!VV,\$TMP1	:CLEAR PORT DEPENDENT BITS FROM THE COPY
013452	023737	001164	001166		CMP	\$TMP0,\$TMP1	:IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
013460	001006				BNE	66\$:BR IF NOT
013462	005737	001164			TST	\$TMP0	:REGISTERS ARE THE SAME: ARE THEY ZERO ?
013466	001037				BNE	68\$:BR IF NOT
013470	104046				EMT	46	
013472	000137	013656			JMP	70\$:BYPASS THE REST OF THE CHECKS
013476	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT	:SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
013504	013737	001226	001234		MOV	PORTB,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013512	113760	001226	000010		MOV B	PORTB,RPCS2(R0)	:SELECT PORT B.
013520	005737	001164			TST	\$TMP0	:SEE IF STATUS EQ 0 FROM PORT A.
013524	001414				BEQ	67\$:BR IF ZERO
013526	013737	001224	001234		MOV	PORTA,PTNBR	:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013534	013737	001172	001126		MOV	\$TMP3,\$BDDAT	: 'BAD DATA' FOR ERROR TYPE OUT
013542	113760	001224	000010		MOV B	PORTA,RPCS2(R0)	:SELECT PORT A.
013550	005737	001166			TST	\$TMP1	:SEE IF STATUS EQ ZERO FROM PORT B.
013554	001004				BNE	68\$:BR IF NOT
013556	012737	177777	001250	67\$:	MOV	#-1,RELERR	:SET 'RELEASE ERROR' INDICATOR
013564	104026				EMT	26	
013566	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	:LOOK FOR BIT FAILURES WHEN RPDS1 READ
013574	013737	001224	001234		MOV	PORTA,PTNBR	:CHANGE PORT NUMBER
013602	042737	100000	001170		BIC	#ATA,\$TMP2	:DON'T CHECK THE ATTN BIT
013610	023737	001124	001170		CMP	\$GDDAT,\$TMP2	:ALL BITS OK ?
013616	001401				BEQ	69\$:BR IF OK FROM PORT A.
013620	104007				EMT	7	
013622	013737	001172	001126	69\$:	MOV	\$TMP3,\$BDDAT	:CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
013630	013737	001226	001234		MOV	PORTB,PTNBR	:CHANGE PORT NUMBER
013636	042737	100000	001172		BIC	#ATA,\$TMP3	:DON'T CHECK THE ATTN BIT
013644	023737	001124	001172		CMP	\$GDDAT,\$TMP3	:SEE IF READ OK FROM PORT B.
013652	001401				BEQ	70\$:BR IF OK
013654	104007				EMT	7	
013656	000240			70\$:	NOP		
013660	005737	001250			TST	RELERR	:DID DRIVE RETURN TO NEUTRAL ?
013664	001402				BEQ	:+6	:BR IF IN NEUTRAL
013666	000137	014142			JMP	1\$:GO WAIT FOR DRIVE TO TIMEOUT
013672	113760	001226	000010		MOV B	PORTB,RPCS2(R0)	:SELECT PORT B
013700	013737	001226	001234		MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013706	005037	001244			CLR	CKERR	:CLEAR THE 'CHECK ERROR' INDICATOR
013712	016037	000012	001126		MOV	RPDS1(R0),\$BDDAT	:GET CONTENTS OF RPDS1
013720	012737	000012	001122		MOV	#RPDS1,\$BDADR	:FORM REGISTER ADDRESS OF ERROR MESSAGE
013726	060037	001122			ADD	R0,\$BDADR	:ADD RHI1 BASE ADDRESS
013732	005037	001124			CLR	\$GDDAT	:WHAT REGISTER SHOULD BE
013736	013737	001126	001164		MOV	\$BDDAT,\$TMP0	:MOVE REGISTER CONTENTS TO '\$TMP0'
013744	042737	077777	001164		BIC	#*CATA,\$TMP0	:SAVE SPECIFIED BITS
013752	023737	001124	001164		CMP	\$GDDAT,\$TMP0	:COMPARE THE BITS
013760	001414				BEQ	71\$:BR IF OK
013762	013737	001126	001174		MOV	\$BDDAT,\$TMP4	:COPY 'BAD DATA'
013770	042737	100000	001174		BIC	#ATA,\$TMP4	:CLEAR THE MASKED BITS
013776	053737	001174	001124		BIS	\$TMP4,\$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT
014004	104017				EMT	17	

```
014006 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
014012 000240          71$:    NOP
014014 113760 001224 000010    MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
014022 013737 001224 001234    MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014030 005037 001244          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
014034 016037 000012 001126    MOV     RPDS1(R0),SBDDAT ;GET CONTENTS OF RPDS1
014042 012737 000012 001122    MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
014050 060037 001122          ADD     R0,$BDADR      ;ADD RHI1 BASE ADDRESS
014054 005037 001124          CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
014060 013737 001126 001164    MOV     SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
014066 042737 077777 001164    BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
014074 023737 001124 001164    CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
014102 001414          BEQ     73$           ;BR IF OK
014104 013737 001126 001174    MOV     SBDDAT,$TMP4  ;COPY 'BAD DATA'
014112 042737 100000 001174    BIC     #ATA,$TMP4    ;CLEAR THE MASKED BITS
014120 053737 001174 001124    BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
014126 104017          EMT     17
014130 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
014134 000240          73$:    NOP
014136 000137 014174          JMP     2$            ;GO CHECK FOR LOOP ON ERROR
```

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

```
014142          1$:
014142 113760 001224 000010    MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
014150 013737 001224 001234    MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014156 005760 000012          TST     RPDS1(R0)      ;WAIT FOR TIMEOUT TO RELEASE DRIVE
014162 001004          BNE     2$            ;BR WHEN DRIVE RELEASED
014164 005737 001254          TST     WATCH         ;CHECK THE WATCH
014170 001364          BNE     1$            ;BR IF NOT ZERO
014172 104036          EMT     36
014174 000004          2$:    SCOPE          ;LOOP ?
```

315
324
325

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

```
014176          TST10:
014176 005737 001274          TST     KYBCTL        ;PERFORMING ONLY SINGLE TESTS ?
014202 001406          BEQ     2$            ;BR IF NOT
014204 100002          BPL     1$            ;BR IF JUST ENTERED TEST
014206 000137 003010          JMP     EXEC          ;RETURN & GET NEXT TEST NUMBER
014212 012737 177777 001274    1$:    MOV     #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
014220 112737 000010 001102    2$:    MOVB    #10,$STNM    ;TEST NUMBER
014226 012737 014250 001106    MOV     #TEST10,$LPADR ;LOAD LOOP ON TEST ADDRESS
014234 012737 014250 001110    MOV     #TEST10,$LPERR ;LOAD LOOP ON ERROR ADDRESS
014242 012737 000144 001176    MOV     #100,$TIMES   ;DO 100. ITERATIONS
326 014250 012706 001100    TEST10: MOV     #STACK,SP    ;LOAD THE STACK POINTER
340 014254 113760 001224 000010    MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
014262 013737 001224 001234    MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
014270 013737 001224 001236      MOV     PORTA,SEIZPT      ;ADDR OF PORT WHICH WILL ISSUE RELEASE
                                ;ISSUE A RELEASE COMMAND

014276 012760 000013 000000      MOV     #13,RPCS1(RO)    ;ISSUE A RELEASE COMMAND
                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

014304 005037 001250              CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
014310 012737 000012 001122      MOV     #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
014316 060037 001122              ADD     RO,$BDADR        ;ADD THE I/O BASE ADDRESS
014322 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
014330 113760 001224 000010      MOV     PORTA,RPCS2(RO)  ;SELECT PORT A.
014336 016037 000012 001170      MOV     RPDS1(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
014344 013737 001170 001164      MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
014352 042737 100100 001164      BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014360 113760 001226 000010      MOV     PORTB,RPCS2(RO)  ;SELECT PORT B.
014366 016037 000012 001172      MOV     RPDS1(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
014374 013737 001172 001166      MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
014402 042737 100100 001166      BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014410 023737 001164 001166      CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
014416 001006                    BNE     64$              ;BR IF NOT
014420 005737 001164              TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
014424 001037                    BNE     66$              ;BR IF NOT
014426 104046                    EMT     46
014430 000137 014614              JMP     68$              ;BYPASS THE REST OF THE CHECKS
014434 013737 001170 001126 64$:  MOV     $TMP2,$GDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014442 013737 001226 001234      MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014450 113760 001226 000010      MOV     PORTB,RPCS2(RO)  ;SELECT PORT B.
014456 005737 001164              TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
014462 001414                    BEQ     65$              ;BR IF ZERO
014464 013737 001224 001234      MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014472 013737 001172 001126      MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
014500 113760 001224 000010      MOV     PORTA,RPCS2(RO)  ;SELECT PORT A.
014506 005737 001166              TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
014512 001004                    BNE     66$              ;BR IF NOT
014514 012737 177777 001250 65$:  MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
014522 104030                    EMT     30
014524 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
014532 013737 001224 001234      MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
014540 042737 100000 001170      BIC     #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
014546 023737 001124 001170      CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
014554 001401                    BEQ     67$              ;BR IF OK FROM PORT A.
014556 104007                    EMT     7
014560 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
014566 013737 001226 001234      MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
014574 042737 100000 001172      BIC     #ATA,$TMP3       ;DON'T CHECK THE ATTN BIT
014602 023737 001124 001172      CMP     $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
014610 001401                    BEQ     68$              ;BR IF OK
014612 104007                    EMT     7
014614 000240 68$:                NOP
014616 000004                    SCOPE                    ;LOOP ?
```

349
350

::*****

```

;*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.
;*
;*****

```

```

014620
014620 005737 001274
014624 001406
014626 100002
014630 000137 003010
014634 012737 177777 001274
014642 112737 000011 001102
014650 012737 014672 001106
014656 012737 014672 001110
014664 012737 000144 001176
351 014672 012706 001100
352 014676 113760 001226 000010
014704 013737 001226 001234
014712 013737 001226 001236

```

```

TST11:
      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      #11,$STNM   ;TEST NUMBER
      MOV      #TEST11,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST11,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #100,$TIMES  ;DO 100. ITERATIONS
TEST11: MOV     #STACK,SP  ;LOAD THE STACK POINTER
      MOV      PORTB,RPCS2(R0) ;SELECT PORT B
      MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      MOV      PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

```

;ISSUE A RELEASE COMMAND

```
014720 012760 000013 000000
```

```
      MOV      #13,RPCS1(R0) ;ISSUE A RELEASE COMMAND
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

014726 005037 001250
014732 012737 000012 001122
014740 060037 001122
014744 012737 011700 001124
014752 113760 001224 000010
014760 016037 000012 001170
014766 013737 001170 001164
014774 042737 100100 001164
015002 113760 001226 000010
015010 016037 000012 001172
015016 013737 001172 001166
015024 042737 100100 001166
015032 023737 001164 001166
015040 001006
015042 005737 001164
015046 001037
015050 104046
015052 000137 015236
015056 013737 001170 001126 64$:
015064 013737 001226 001234
015072 113760 001226 000010
015100 005737 001164
015104 001414
015106 013737 001224 001234
015114 013737 001172 001126

```

```

CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
MOV      #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD      R0,$BDADR       ;ADD THE I/O BASE ADDRESS
MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
MOV      RPDS1(R0),$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(R0) ;SELECT PORT B.
MOV      RPDS1(R0),$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
EMT      46
JMP      68$             ;BYPASS THE REST OF THE CHECKS
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(R0) ;SELECT PORT B.
TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
BEQ      65$             ;BR IF ZERO
MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT

```

```

015122 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
015130 005737 001166              TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
015134 001004              BNE    66$            ;BR IF NOT
015136 012737 177777 001250 65$:  MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
015144 104030              EMT    30
015146 013737 001170 001126 66$:  MOV    $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
015154 013737 001224 001234      MOV    PORTA,PTNBR    ;CHANGE PORT NUMBER
015162 042737 100000 001170      BIC    #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
015170 023737 001124 001170      CMP    $GDDAT,$TMP2   ;ALL BITS OK ?
015176 001401              BEQ    67$            ;BR IF OK FROM PORT A.
015200 104007              EMT    7
015202 013737 001172 001126 67$:  MOV    $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
015210 013737 001226 001234      MOV    PORTB,PTNBR    ;CHANGE PORT NUMBER
015216 042737 100000 001172      BIC    #ATA,$TMP3     ;DON'T CHECK THE ATTN BIT
015224 023737 001124 001172      CMP    $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
015232 001401              BEQ    68$            ;BR IF OK
015234 104007              EMT    7
015236 000240 68$:      NOP
015240 000004              SCOPE                  ;LOOP ?

```

353
375
376

```

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

```

```

015242 005737 001274      TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
015242 001406              BEQ    2$              ;BR IF NOT
015250 100002              BPL    1$              ;BR IF JUST ENTERED TEST
015252 000137 003010      JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
015256 012737 177777 001274 1$:  MOV    #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
015264 112737 000012 001102 2$:  MOVB   #12,$TSTNM     ;TEST NUMBER
015272 012737 015314 001106      MOV    #TEST12,$LPADR ;LOAD LOOP ON TEST ADDRESS
015300 012737 015314 001110      MOV    #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
015306 012737 007640 001176      MOV    #4000,$TIMES   ;DO 4000. ITERATIONS
377 015314 012706 001100  TEST12: MOV    #STACK,$P    ;LOAD THE STACK POINTER
406

```

;SEIZE THE DRIVE THROUGH PORT A

```

015320 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
015326 013737 001224 001236      MOV    PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS

```

```

015334 005060 000012          CLR      RPDS1(R0)      ;WRITE RPDS1
015340 113760 001226 000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
015346 013737 001226 001234  MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015354 013737 001226 001240  MOV     PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
015362 016037 000012 001126  MOV     RPDS1(R0),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
015370 010037 001122          MOV     R0,$BDADR      ;RH11 BASE ADDRESS
015374 062737 000012 001122  ADD     #RPDS1,$BDADR  ;GENERATE BAD REGISTER ADDRESS
015402 005037 001124          CLR     $GDDAT        ;REGISTER SHOULD BE ZERO
015406 023737 001124 001126  CMP     $GDDAT,$BDDAT  ;IS THE REGISTER ZERO
015414 001403          BEQ    64$           ;BR IF IT IS
015416 104004          EMT    4
015420 000137 016616          JMP    1$           ;BYPASS REST OF THE SUBTEST
015424          64$:
015424 113760 001224 000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
015432 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015440 016037 000012 001126  MOV     RPDS1(R0),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
015446 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
015454 013737 001124 001166  MOV     $GDDAT,$STMP1  ;USE GOOD DATA AS A MASK
015462 005137 001166          COM    $STMP1        ;COMPLEMENT THE EXPECTED STATUS
015466 013737 001126 001164  MOV     $BDDAT,$STMP0  ;SAVE THE ACTUAL STATUS
015474 043737 001166 001164  BIC    $STMP1,$STMP0  ;CLEAR UNWANTED BITS
015502 023737 001124 001164  CMP     $GDDAT,$STMP0  ;ARE THE EXPECTED STATUS BITS SET ?
015510 001401          BEQ    65$           ;BR IF THEY ARE
015512 104005          EMT    5
015514 000240          65$:  NOP

;DRIVE CLEAR THROUGH PORT A FIRST

015516 012760 000011 000000  MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR THROUGH PORT A

;VERIFY THAT DRIVE STILL SEIZED BY PORT A

015524 113760 001226 000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
015532 013737 001226 001234  MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015540 005037 001244          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
015544 016037 000012 001126  MOV     RPDS1(R0),SBDDAT ;GET CONTENTS OF RPDS1
015552 012737 000012 001122  MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015560 060037 001122          ADD    R0,$BDADR     ;ADD RH11 BASE ADDRESS
015564 005037 001124          CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
015570 013737 001126 001164  MOV     $BDDAT,$STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
015576 042737 100000 001164  BIC    #^C7777,$STMP0 ;SAVE SPECIFIED BITS
015604 023737 001124 001164  CMP     $GDDAT,$STMP0  ;COMPARE THE BITS
015612 001414          BEQ    66$           ;BR IF OK
015614 013737 001126 001174  MOV     $BDDAT,$STMP4  ;COPY 'BAD DATA'
015622 042737 077777 001174  BIC    #7777,$STMP4   ;CLEAR THE MASKED BITS
015630 053737 001174 001124  BIS    $STMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
015636 104033          EMT    33
015640 005137 001244          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
015644 000240          66$:  NOP
015646 113760 001224 000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
015654 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015662 005037 001244          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
015666 016037 000012 001126  MOV     RPDS1(R0),SBDDAT ;GET CONTENTS OF RPDS1
015674 012737 000012 001122  MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015702 060037 001122          ADD    R0,$BDADR     ;ADD RH11 BASE ADDRESS
015706 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
015714 013737 001126 001164  MOV     $BDDAT,$STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
  
```

```

015722 042737 100000 001164      BIC    #^C77777,$STMP0  ;SAVE SPECIFIED BITS
015730 023737 001124 001164      CMP    $GDDAT,$STMP0   ;COMPARE THE BITS
015736 001414                      BEQ    68$              ;BR IF OK
015740 013737 001126 001174      MOV    $BDDAT,$STMP4   ;COPY 'BAD DATA'
015746 042737 077777 001174      BIC    #77777,$STMP4   ;CLEAR THE MASKED BITS
015754 053737 001174 001124      BIS    $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
015762 104033                      EMT    33
015764 005137 001244                      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
015770 000240                      NOP
  
```

68\$:

;NOW ISSUE MASSBUS INIT

```

015772 012760 000040 000010      MOV    #CLR,RPCS2(R0) ;ISSUE MASSBUS INIT
  
```

;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

```

016000 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
016006 013737 001226 001234      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016014 005037 001244                      CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
016020 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
016026 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016034 060037 001122                      ADD    R0,$BDADR      ;ADD RH11 BASE ADDRESS
016040 005037 001124                      CLR    $GDDAT         ;WHAT REGISTER SHOULD BE
016044 013737 001126 001164      MOV    $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
016052 042737 100000 001164      BIC    #^C77777,$STMP0 ;SAVE SPECIFIED BITS
016060 023737 001124 001164      CMP    $GDDAT,$STMP0   ;COMPARE THE BITS
016066 001414                      BEQ    70$              ;BR IF OK
016070 013737 001126 001174      MOV    $BDDAT,$STMP4   ;COPY 'BAD DATA'
016076 042737 077777 001174      BIC    #77777,$STMP4   ;CLEAR THE MASKED BITS
016104 053737 001174 001124      BIS    $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
016112 104034                      EMT    34
016114 005137 001244                      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
016120 000240                      NOP
  
```

70\$:

```

016122 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
016130 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016136 005037 001244                      CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
016142 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
016150 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016156 060037 001122                      ADD    R0,$BDADR      ;ADD RH11 BASE ADDRESS
016162 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
016170 013737 001126 001164      MOV    $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
016176 042737 100000 001164      BIC    #^C77777,$STMP0 ;SAVE SPECIFIED BITS
016204 023737 001124 001164      CMP    $GDDAT,$STMP0   ;COMPARE THE BITS
016212 001414                      BEQ    72$              ;BR IF OK
016214 013737 001126 001174      MOV    $BDDAT,$STMP4   ;COPY 'BAD DATA'
016222 042737 077777 001174      BIC    #77777,$STMP4   ;CLEAR THE MASKED BITS
016230 053737 001174 001124      BIS    $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
016236 104034                      EMT    34
016240 005137 001244                      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
016244 000240                      NOP
  
```

72\$:

;RELEASE THE DRIVE FROM PORT A

```

016246 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
016254 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016262 012760 000013 000000      MOV    #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT A
  
```


:VERIFY THAT THE DRIVE IS IN NEUTRAL

```

016270 005037 001250          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
016274 012737 000012 001122  MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
016302 060037 001122          ADD      R0,$BDADR   ;ADD THE I/O BASE ADDRESS
016306 012737 011700 001124  MOV      #MCL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
016314 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
016322 016037 000012 001170  MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
016330 013737 001170 001164  MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
016336 042737 100100 001164  BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016344 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
016352 016037 000012 001172  MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
016360 013737 001172 001166  MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
016366 042737 100100 001166  BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016374 023737 001164 001166  CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
016402 001006          BNE      74$          ;BR IF NOT
016404 005737 001164          TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
016410 001045          BNE      76$          ;BR IF NOT
016412 104046          EMT      46
016414 000137 016614          JMP      78$          ;BYPASS THE REST OF THE CHECKS
016420 013737 001170 001126 74$: MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
016426 013737 001226 001234  MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016434 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
016442 005737 001164          TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
016446 001414          BEQ      75$          ;BR IF ZERO
016450 013737 001224 001234  MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016456 013737 001172 001126  MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
016464 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
016472 005737 001166          TST      $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
016476 001012          BNE      76$          ;BR IF NOT
016500 012737 177777 001250 75$: MOV      #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
016506 012760 000011 000000  MOV      #11,RPCS1(R0) ;CLEAR THE DRIVE
016514 012760 000013 000000  MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
016522 104026          EMT      26
016524 013737 001170 001126 76$: MOV      $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
016532 013737 001224 001234  MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
016540 042737 100000 001170  BIC      #ATA,$TMP2   ;DON'T CHECK THE ATTN BIT
016546 023737 001124 001170  CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
016554 001401          BEQ      77$          ;BR IF OK FROM PORT A.
016556 104007          EMT      7
016560 013737 001172 001126 77$: MOV      $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
016566 013737 001226 001234  MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
016574 042737 100000 001172  BIC      #ATA,$TMP3   ;DON'T CHECK THE ATTN BIT
016602 023737 001124 001172  CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
016610 001401          BEQ      78$          ;BR IF OK
016612 104007          EMT      7
016614 000240          78$: NOP
016616 000004          1$: SCOPE          ;LOOP ?
  
```

425
426

```

:*****
:*TEST 13      TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
:*
:*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
:*      PORT TO RELEASE THE DRIVE.
:*
:*      A.  SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.
:*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
  
```

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

 TST13:

016620
 016620 005737 001274
 016624 001406
 016626 100002
 016630 000137 003010
 016634 012737 177777 001274
 016642 112737 000013 001102
 016650 012737 016672 001106
 016656 012737 016672 001110
 016664 012737 007640 001176
 427 016672 012706 001100
 428

```

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 #13,$STNM ;TEST NUMBER
MOV #TEST13,$LPADR ;LOAD LOOP ON TEST ADDRESS
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

016676 113760 001226 000010
 016704 013737 001226 001236
 016712 005060 000012
 016716 113760 001224 000010
 016724 013737 001224 001234
 016732 013737 001224 001240
 016740 016037 000012 001126
 016746 010037 001122
 016752 062737 000012 001122
 016760 005037 001124
 016764 023737 001124 001126
 016772 001403
 016774 104004
 016776 000137 020174
 017002
 017002 113760 001226 000010
 017010 013737 001226 001234
 017016 016037 000012 001126
 017024 012737 011700 001124
 017032 013737 001124 001166
 017040 005137 001166
 017044 013737 001126 001164
 017052 043737 001166 001164
 017060 023737 001124 001164
 017066 001401
 017070 104005
 017072 000240

```

MOV PORTB,RPCS2(R0) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(R0) ;WRITE RPDS1
MOVB PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$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
EMT 4
JMP 1$ ;BYPASS REST OF THE SUBTEST
64$: MOVB PORTB,RPCS2(R0) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPDS1(R0),$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
EMT 5
65$: NOP
  
```

;DRIVE CLEAR THROUGH PORT B FIRST

017074 012760 000011 000000

```

MOV #11,RPCS1(R0) ;ISSUE DRIVE CLEAR THROUGH PORT B
  
```

;VERIFY THAT DRIVE STILL SEIZED BY PORT B

017102	113760	001224	000010	MOV	PORTA,RPCS2(R0)	;SELECT PORT A
017110	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017116	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
017122	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
017130	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
017136	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
017142	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
017146	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
017154	042737	100000	001164	BIC	#^C77777,\$TMP0	;SAVE SPECIFIED BITS
017162	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
017170	001414			BEQ	66\$;BR IF OK
017172	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
017200	042737	077777	001174	BIC	#77777,\$TMP4	;CLEAR THE MASKED BITS
017206	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
017214	104033			EMT	33	
017216	005137	001244		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
017222	000240			66\$: NOP		
017224	113760	001226	000010	MOV	PORTB,RPCS2(R0)	;SELECT PORT B
017232	013737	001226	001234	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017240	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
017244	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
017252	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
017260	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
017264	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;WHAT REGISTER SHOULD BE
017272	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
017300	042737	100000	001164	BIC	#^C77777,\$TMP0	;SAVE SPECIFIED BITS
017306	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
017314	001414			BEQ	68\$;BR IF OK
017316	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
017324	042737	077777	001174	BIC	#77777,\$TMP4	;CLEAR THE MASKED BITS
017332	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
017340	104033			EMT	33	
017342	005137	001244		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
017346	000240			68\$: NOP		

;NOW ISSUE MASSBUS INIT

017350	012760	000040	000010	MOV	#CLR,RPCS2(R0)	;ISSUE MASSBUS INIT
--------	--------	--------	--------	-----	----------------	---------------------

;CONFIRM THAT DRIVE STILL SEIZED BY PORT B

017356	113760	001224	000010	MOV	PORTA,RPCS2(R0)	;SELECT PORT A
017364	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017372	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
017376	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
017404	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
017412	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
017416	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
017422	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
017430	042737	100000	001164	BIC	#^C77777,\$TMP0	;SAVE SPECIFIED BITS
017436	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
017444	001414			BEQ	70\$;BR IF OK
017446	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
017454	042737	077777	001174	BIC	#77777,\$TMP4	;CLEAR THE MASKED BITS

```

017462 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
017470 104034                      EMT      34
017472 005137 001244              COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
017476 000240                      NOP
70$: 017500 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B
017506 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017514 005037 001244              CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
017520 016037 000012 001126      MOV      RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
017526 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017534 060037 001122              ADD      R0,$BDADR        ;ADD RH11 BASE ADDRESS
017540 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
017546 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
017554 042737 100000 001164      BIC      #^C7777,$TMP0    ;SAVE SPECIFIED BITS
017562 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
017570 001414                      BEQ      72$              ;BR IF OK
017572 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
017600 042737 077777 001174      BIC      #77777,$TMP4     ;CLEAR THE MASKED BITS
017606 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
017614 104034                      EMT      34
017616 005137 001244              COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
017622 000240                      NOP

                                ;RELEASE THE DRIVE FROM PORT B

017624 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B
017632 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017640 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

017646 005037 001250              CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
017652 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
017660 060037 001122              ADD      R0,$BDADR        ;ADD THE I/O BASE ADDRESS
017664 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
017672 113760 001224 000010      MOV      PORTA,RPCS2(R0)   ;SELECT PORT A.
017700 016037 000012 001170      MOV      RPDS1(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
017706 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
017714 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
017722 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B.
017730 016037 000012 001172      MOV      RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
017736 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
017744 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
017752 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
017760 001006                      BNE      74$              ;BR IF NOT
017762 005737 001164              TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
017766 001045                      BNE      76$              ;BR IF NOT
017770 104046                      EMT      46
017772 000137 020172              JMP      78$              ;BYPASS THE REST OF THE CHECKS
017776 013737 001170 001126 74$: MOV      $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
020004 013737 001226 001234      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020012 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B.
020020 005737 001164              TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
020024 001414                      BEQ      75$              ;BR IF ZERO
020026 013737 001224 001234      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020034 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
020042 113760 001224 000010      MOV      PORTA,RPCS2(R0)   ;SELECT PORT A.
020050 005737 001166              TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
  
```

```

020054 001012          BNE      76$          :BR IF NOT
020056 012737 177777 001250 75$:  MOV     #-1,RELERR    :SET 'RELEASE ERROR' INDICATOR
020064 012760 000011 000000      MOV     #11,RPCS1(R0) :CLEAR THE DRIVE
020072 012760 000013 000000      MOV     #13,RPCS1(R0) :RELEASE THE DRIVE
020100 104026          EMT      26
020102 013737 001170 001126 76$:  MOV     $TMP2,$BDDAT   :LOOK FOR BIT FAILURES WHEN RPDS1 READ
020110 013737 001224 001234      MOV     PORTA,PTNBR    :CHANGE PORT NUMBER
020116 042737 100000 001170      BIC     #ATA,$TMP2     :DON'T CHECK THE ATTN BIT
020124 023737 001124 001170      CMP     $GDDAT,$TMP2   :ALL BITS OK ?
020132 001401          BEQ     77$          :BR IF OK FROM PORT A.
020134 104007          EMT      7
020136 013737 001172 001126 77$:  MOV     $TMP3,$BDDAT   :CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
020144 013737 001226 001234      MOV     PORTB,PTNBR    :CHANGE PORT NUMBER
020152 042737 100000 001172      BIC     #ATA,$TMP3     :DON'T CHECK THE ATTN BIT
020160 023737 001124 001172      CMP     $GDDAT,$TMP3   :SEE IF READ OK FROM PORT B.
020166 001401          BEQ     78$          :BR IF OK
020170 104007          EMT      7
020172 000240          NOP
020174 000004          1$:  SCOPE          :LOOP ?
  
```

429
448
449

```

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

450
483

```

020176 005737 001274          TST14: TST     KYBCTL      :PERFORMING ONLY SINGLE TESTS ?
020176 001406          BEQ     2$          :BR IF NOT
020202 100002          BPL     1$          :BR IF JUST ENTERED TEST
020206 000137 003010          JMP     EXEC        :RETURN & GET NEXT TEST NUMBER
020212 012737 177777 001274 1$:  MOV     #-1,KYBCTL    :SET SINGLE TEST INDICATOR
020220 112737 000014 001102 2$:  MOV     #14,$STNM    :TEST NUMBER
020226 012737 020250 001106      MOV     #TEST14,$LPADR :LOAD LOOP ON TEST ADDRESS
020234 012737 020250 001110      MOV     #TEST14,$LPERR :LOAD LOOP ON ERROR ADDRESS
020242 012737 000004 001176      MOV     #4,$TIMES    :DO 4 ITERATIONS
020250 012706 001100          TEST14: MOV     #STACK,SP   :LOAD THE STACK POINTER

          :SET ATTENTION BITS FOR BOTH PORTS
020254 113760 001224 000010      MOV     PORTA,RPCS2(R0) :SELECT PORT 64$
020262 012760 177777 000014      MOV     #-1,RPER1(R0)  :FORCE ERRORS
020270 005060 000014          CLR     RPER1(R0)    :CLEAR THE ERRORS
  
```

```

020274 013760 001226 000010      MOV    PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
020302 005760 000012      64$:  TST    RPDS1(R0)    ;WAIT FOR DRIVE TO TIMEOUT
020306 001775                BEQ    64$           ;BR IF DRIVE HASN'T TIMED OUT
020310 012760 177777 000014      MOV    #-1,RPER1(R0) ;FORCE ERRORS ON PORT 65$
020316 005060 000014      CLR    RPER1(R0)    ;CLEAR THE ERRORS
020322 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT '64$' AGAIN
020330 005760 000012      65$:  TST    RPDS1(R0)    ;WAIT FOR DRIVE TO TIMEOUT
020334 001775                BEQ    65$           ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

020336 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
020344 013737 001224 001234      MOV    PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020352 005037 001244                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
020356 016037 000012 001126      MOV    RPDS1(R0),%SDDAT ;GET CONTENTS OF RPDS1
020364 012737 000012 001122      MOV    %RPDS1,%SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020372 060037 001122                ADD    R0,%SBDADR   ;ADD RH11 BASE ADDRESS
020376 012737 100000 001124      MOV    %ATA,%SGDDAT ;WHAT REGISTER SHOULD BE
020404 013737 001126 001164      MOV    %SDDAT,%STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
020412 042737 077777 001164      BIC    %^CATA,%STMP0 ;SAVE SPECIFIED BITS
020420 023737 001124 001164      CMP    %SGDDAT,%STMP0 ;COMPARE THE BITS
020426 001414                BEQ    66$           ;BR IF OK
020430 013737 001126 001174      MOV    %SDDAT,%STMP4 ;COPY 'BAD DATA'
020436 042737 100000 001174      BIC    %ATA,%STMP4  ;CLEAR THE MASKED BITS
020444 053737 001174 001124      BIS    %STMP4,%SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
020452 104010                EMT    10
020454 005137 001244                COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
020460 000240      66$:  NOP
020462 005737 001244                TST    CKERR        ;WAS ATTN BIT FOR PORT A SET ?
020466 001402                BEQ    .+6           ;BR IF IT WAS
020470 000137 021476                JMP    1$           ;BYPASS REST OF TEST IF NOT
020474 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
020502 013737 001226 001234      MOV    PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020510 005037 001244                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
020514 016037 000012 001126      MOV    RPDS1(R0),%SDDAT ;GET CONTENTS OF RPDS1
020522 012737 000012 001122      MOV    %RPDS1,%SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020530 060037 001122                ADD    R0,%SBDADR   ;ADD RH11 BASE ADDRESS
020534 012737 100000 001124      MOV    %ATA,%SGDDAT ;WHAT REGISTER SHOULD BE
020542 013737 001126 001164      MOV    %SDDAT,%STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
020550 042737 077777 001164      BIC    %^CATA,%STMP0 ;SAVE SPECIFIED BITS
020556 023737 001124 001164      CMP    %SGDDAT,%STMP0 ;COMPARE THE BITS
020564 001414                BEQ    68$           ;BR IF OK
020566 013737 001126 001174      MOV    %SDDAT,%STMP4 ;COPY 'BAD DATA'
020574 042737 100000 001174      BIC    %ATA,%STMP4  ;CLEAR THE MASKED BITS
020602 053737 001174 001124      BIS    %STMP4,%SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
020610 104010                EMT    10
020612 005137 001244                COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
020616 000240      68$:  NOP
020620 005737 001244                TST    CKERR        ;WAS ATTN BIT FOR PORT B SET ?
020624 001402                BEQ    .+6           ;BR IF IT WAS
020626 000137 021476                JMP    1$           ;BYPASS REST OF TEST IF NOT
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

020632 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
020640 013737 001224 001236      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
  
```

```

020646 005060 000012          CLR      RPDS1(R0)      ;WRITE RPDS1
020652 013737 001226 001240  MOV      PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS

;ISSUE MASSBUS INIT TO PORT A

020660 012760 000040 000010    MOV      #CLR,RPCS2(R0) ;MASSBUS INIT
020666 113760 001224 000010    MOV      PORTA,RPCS2(R0) ;SELECT PORT A AGAIN

;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED

020674 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
020700 016037 000012 001126    MOV      RPDS1(R0),%BDDAT ;GET CONTENTS OF RPDS1
020706 012737 000012 001122    MOV      #RPDS1,%BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020714 060037 001122          ADD      R0,%BDADR      ;ADD RH11 BASE ADDRESS
020720 005037 001124          CLR      %GDDAT        ;WHAT REGISTER SHOULD BE
020724 013737 001126 001164    MOV      %BDDAT,%STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
020732 042737 077777 001164    BIC      #^CATA,%STMP0 ;SAVE SPECIFIED BITS
020740 023737 001124 001164    CMP      %GDDAT,%STMP0 ;COMPARE THE BITS
020746 001414          BEQ      72$           ;BR IF OK
020750 013737 001126 001174    MOV      %BDDAT,%STMP4 ;COPY 'BAD DATA'
020756 042737 100000 001174    BIC      #ATA,%STMP4   ;CLEAR THE MASKED BITS
020764 053737 001174 001124    BIS      %STMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
020772 104047          EMT      47
020774 005137 001244          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
021000 000240          NOP

72$:

;RELEASE THE DRIVE FROM PORT A

021002 113760 001224 000010    MOV      PORTA,RPCS2(R0) ;SELECT PORT A
021010 013737 001224 001234    MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021016 012760 000013 000000    MOV      #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

021024 005037 001250          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
021030 012737 000012 001122    MOV      #RPDS1,%BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
021036 060037 001122          ADD      R0,%BDADR      ;ADD THE I/O BASE ADDRESS
021042 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV,%GDDAT ;COMPARISON CONSTANT
021050 113760 001224 000010    MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
021056 016037 000012 001170    MOV      RPDS1(R0),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
021064 013737 001170 001164    MOV      %STMP2,%STMP0  ;COPY IT INTO 'STMP0'
021072 042737 100100 001164    BIC      #ATA!VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021100 113760 001226 000010    MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
021106 016037 000012 001172    MOV      RPDS1(R0),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
021114 013737 001172 001166    MOV      %STMP3,%STMP1 ;COPY IT INTO 'STMP1'
021122 042737 100100 001166    BIC      #ATA!VV,%STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021130 023737 001164 001166    CMP      %STMP0,%STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
021136 001006          BNE      74$          ;BR IF NOT
021140 005737 001164          TST      %STMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
021144 001045          BNE      76$          ;BR IF NOT
021146 104046          EMT      46
021150 000137 021350          JMP      78$          ;BYPASS THE REST OF THE CHECKS
021154 013737 001170 001126 74$: MOV      %STMP2,%BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
021162 013737 001226 001234    MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021170 113760 001226 000010    MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
021176 005737 001164          TST      %STMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
  
```

```

021202 001414          BEQ      75$      ;BR IF ZERO
021204 013737 001224 001234  MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021212 013737 001172 001126  MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
021220 113760 001224 000010  MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
021226 005737 001166          TST     $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
021232 001012          BNE     76$      ;BR IF NOT
021234 012737 177777 001250 75$:  MOV     #-1,RELERR  ;SET 'RELEASE ERROR' INDICATOR
021242 012760 000011 000000  MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
021250 012760 000013 000000  MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
021256 104026          EMT     26
021260 013737 001170 001126 76$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
021266 013737 001224 001234  MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
021274 042737 100000 001170  BIC     #ATA,$TMP2  ;DON'T CHECK THE ATTN BIT
021302 023737 001124 001170  CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
021310 001401          BEQ     77$      ;BR IF OK FROM PORT A.
021312 104007          EMT     7
021314 013737 001172 001126 77$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
021322 013737 001226 001234  MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
021330 042737 100000 001172  BIC     #ATA,$TMP3  ;DON'T CHECK THE ATTN BIT
021336 023737 001124 001172  CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
021344 001401          BEQ     78$      ;BR IF OK
021346 104007          EMT     7
021350 000240          78$:  NOP

```

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

```

021352 113760 001226 000010  MOVVB  PORTB,RPCS2(R0) ;SELECT PORT B
021360 013737 001226 001234  MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021366 005037 001244          CLR     CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
021372 016037 000012 001126  MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
021400 012737 000012 001122  MOV     #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021406 060037 001122          ADD     R0,$BDADR  ;ADD RH11 BASE ADDRESS
021412 012737 100000 001124  MOV     #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
021420 013737 001126 001164  MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
021426 042737 077777 001164  BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
021434 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
021442 001414          BEQ     79$      ;BR IF OK
021444 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
021452 042737 100000 001174  BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
021460 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
021466 104050          EMT     50
021470 005137 001244          COM     CKERR    ;SET THE REGISTER COMPARE ERROR INDICATOR
021474 000240          79$:  NOP
021476 000004          1$:  SCOPE ;LOOP ?

```

501
502

```

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

```

503 021500      021500      005737      001274
504 021504      021504      001406
021506      100002
021510      000137      003010
021514      012737      177777      001274      1$:
021522      112737      000015      001102      2$:
021530      012737      021552      001106
021536      012737      021552      001110
021544      012737      000004      001176
021552      012706      001100      TEST15:

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

      ;SET ATTENTION BITS FOR BOTH PORTS

021556      113760      001224      000010
021564      012760      177777      000014
021572      005060      000014
021576      013760      001226      000010
021604      005760      000012      64$:
021610      001775
021612      012760      177777      000014
021620      005060      000014
021624      113760      001224      000010
021632      005760      000012      65$:
021636      001775

      MOVB     PORTA,$RPCS2(R0) ;SELECT PORT 64$
      MOV      #-1,$RPER1(R0)  ;FORCE ERRORS
      CLR      $RPER1(R0)      ;CLEAR THE ERRORS
      MOV      PORTB,$RPCS2(R0) ;SELECT THE OTHER PORT
64$:   TST      $RPDS1(R0)      ;WAIT FOR DRIVE TO TIMEOUT
      BEQ      64$            ;BR IF DRIVE HASN'T TIMED OUT
      MOV      #-1,$RPER1(R0)  ;FORCE ERRORS ON PORT 65$
      CLR      $RPER1(R0)      ;CLEAR THE ERRORS
      MOVB     PORTA,$RPCS2(R0) ;SELECT PORT '64$' AGAIN
65$:   TST      $RPDS1(R0)      ;WAIT FOR DRIVE TO TIMEOUT
      BEQ      65$            ;BR IF DRIVE HASN'T TIMED OUT

      ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

021640      113760      001226      000010
021646      013737      001226      001234
021654      005037      001244
021660      016037      000012      001126
021666      012737      000012      001122
021674      060037      001122
021700      012737      100000      001124
021706      013737      001126      001164
021714      042737      077777      001164
021722      023737      001124      001164
021730      001414
021732      013737      001126      001174
021740      042737      100000      001174
021746      053737      001174      001124
021754      104010
021756      005137      001244
021762      000240      66$:
021764      005737      001244
021770      001402
021772      000137      023000
021776      113760      001224      000010
022004      013737      001224      001234
022012      005037      001244

      MOVB     PORTB,$RPCS2(R0) ;SELECT PORT B
      MOV      PORTB,$PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR      $CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
      MOV      $RPDS1(R0),$SBDAT ;GET CONTENTS OF RPDS1
      MOV      #R$RPDS1,$SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      ADD      R0,$SBDADR      ;ADD RH11 BASE ADDRESS
      MOV      #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
      MOV      $SBDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
      BIC      #^CATA,$STMP0   ;SAVE SPECIFIED BITS
      CMP      $GDDAT,$STMP0   ;COMPARE THE BITS
      BEQ      66$            ;BR IF OK
      MOV      $SBDAT,$STMP4   ;COPY 'BAD DATA'
      BIC      #ATA,$STMP4     ;CLEAR THE MASKED BITS
      BIS      $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
      EMT      10
      COM      $CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
66$:   NOP
      TST      $CKERR          ;WAS ATTN BIT FOR PORT B SET ?
      BEQ      +6            ;BR IF IT WAS
      JMP      1$            ;BYPASS REST OF TEST IF NOT
      MOVB     PORTA,$RPCS2(R0) ;SELECT PORT A
      MOV      PORTA,$PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      CLR      $CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
  
```

```

022016 016037 000012 001126      MOV      RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
022024 012737 000012 001122      MOV      #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022032 060037 001122                ADD      R0, $BDADR      ;ADD RH11 BASE ADDRESS
022036 012737 100000 001124      MOV      #ATA, $GDDAT   ;WHAT REGISTER SHOULD BE
022044 013737 001126 001164      MOV      $BDDAT, $TMPO  ;MOVE REGISTER CONTENTS TO '$TMPO'
022052 042737 077777 001164      BIC      #^CATA, $TMPO  ;SAVE SPECIFIED BITS
022060 023737 001124 001164      CMP      $GDDAT, $TMPO  ;COMPARE THE BITS
022066 001414                BEQ      68$            ;BR IF OK
022070 013737 001126 001174      MOV      $BDDAT, $TMP4  ;COPY 'BAD DATA'
022076 042737 100000 001174      BIC      #ATA, $TMP4    ;CLEAR THE MASKED BITS
022104 053737 001174 001124      BIS      $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
022112 104010                EMT      10
022114 005137 001244                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
022120 000240                NOP
022122 005737 001244                TST      CKERR          ;WAS ATTN BIT FOR PORT A SET ?
022126 001402                BEQ      +6            ;BR IF IT WAS
022130 000137 023000                JMP      1$            ;BYPASS REST OF TEST IF NOT
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

022134 113760 001226 000010      MOV      PORTB, RPCS2(R0) ;SELECT PORT B
022142 013737 001226 001236      MOV      PORTB, SEIZPT  ;STORE SEIZING PORT'S ADDRESS
022150 005060 000012                CLR      RPDS1(R0)      ;WRITE RPDS1
022154 013737 001224 001240      MOV      PORTA, OPPRT   ;'OPPOSITE' PORT ADDRESS
  
```

;ISSUE MASSBUS INIT TO PORT B

```

022162 012760 000040 000010      MOV      #CLR, RPCS2(R0) ;MASSBUS INIT
022170 113760 001226 000010      MOV      PORTB, RPCS2(R0) ;SELECT PORT B AGAIN
  
```

;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED

```

022176 005037 001244                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
022202 016037 000012 001126      MOV      RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
022210 012737 000012 001122      MOV      #RPDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022216 060037 001122                ADD      R0, $BDADR      ;ADD RH11 BASE ADDRESS
022222 005037 001124                CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
022226 013737 001126 001164      MOV      $BDDAT, $TMPO  ;MOVE REGISTER CONTENTS TO '$TMPO'
022234 042737 077777 001164      BIC      #^CATA, $TMPO  ;SAVE SPECIFIED BITS
022242 023737 001124 001164      CMP      $GDDAT, $TMPO  ;COMPARE THE BITS
022250 001414                BEQ      72$            ;BR IF OK
022252 013737 001126 001174      MOV      $BDDAT, $TMP4  ;COPY 'BAD DATA'
022260 042737 100000 001174      BIC      #ATA, $TMP4    ;CLEAR THE MASKED BITS
022266 053737 001174 001124      BIS      $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
022274 104047                EMT      47
022276 005137 001244                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
022302 000240                NOP
  
```

;RELEASE THE DRIVE FROM PORT B

```

022304 113760 001226 000010      MOV      PORTB, RPCS2(R0) ;SELECT PORT B
022312 013737 001226 001234      MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022320 012760 000013 000000      MOV      #13, RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

022326 005037 001250          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
022332 012737 000012 001122  MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
022340 060037 001122          ADD      R0,$BDADR   ;ADD THE I/O BASE ADDRESS
022344 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
022352 113760 001224 000010  MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
022360 016037 000012 001170  MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022366 013737 001170 001164  MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
022374 042737 100100 001164  BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022402 113760 001226 000010  MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
022410 016037 000012 001172  MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022416 013737 001172 001166  MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
022424 042737 100100 001166  BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022432 023737 001164 001166  CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
022440 001006          BNE      74$          ;BR IF NOT
022442 005737 001164          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
022446 001045          BNE      76$          ;BR IF NOT
022450 104046          EMT      46
022452 000137 022652          JMP      78$          ;BYPASS THE REST OF THE CHECKS
022456 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
022464 013737 001226 001234  MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
022472 113760 001226 000010  MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
022500 005737 001164          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
022504 001414          BEQ      75$          ;BR IF ZERO
022506 013737 001224 001234  MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
022514 013737 001172 001126  MOV      $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
022522 113760 001224 000010  MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
022530 005737 001166          TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
022534 001012          BNE      76$          ;BR IF NOT
022536 012737 177777 001250 75$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
022544 012760 000011 000000  MOV      #11,RPCS1(R0) ;CLEAR THE DRIVE
022552 012760 000013 000000  MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
022560 104026          EMT      26
022562 013737 001170 001126 76$:  MOV      $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
022570 013737 001224 001234  MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
022576 042737 100000 001170  BIC      #ATA,$TMP2    ;DON'T CHECK THE ATTN BIT
022604 023737 001124 001170  CMP      $GDDAT,$TMP2  ;ALL BITS OK ?
022612 001401          BEQ      77$          ;BR IF OK FROM PORT A.
022614 104007          EMT      7
022616 013737 001172 001126 77$:  MOV      $TMP3,$BDDAT  ;CHECK RPDS1 FOR BIT FAILURES - FROM PCRT B.
022624 013737 001226 001234  MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
022632 042737 100000 001172  BIC      #ATA,$TMP3    ;DON'T CHECK THE ATTN BIT
022640 023737 001124 001172  CMP      $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.
022646 001401          BEQ      78$          ;BR IF OK
022650 104007          EMT      7
022652 000240          78$:  NOP

```

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

```

022654 113760 001224 000010  MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
022662 013737 001224 001234  MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022670 005037 001244          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
022674 016037 000012 001126  MOV      RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
022702 012737 000012 001122  MOV      #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022710 060037 001122          ADD      R0,$BDADR   ;ADD RH11 BASE ADDRESS
022714 012737 100000 001124  MOV      #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
022722 013737 001126 001164  MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'

```

```

022730 042737 077777 001164      BIC    #^CATA,$TMP0 ;SAVE SPECIFIED BITS
022736 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
022744 001414                      BEQ    79$           ;BR IF OK
022746 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
022754 042737 100000 001174      BIC    #ATA,$TMP4   ;CLEAR THE MASKED BITS
022762 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
022770 104050                      EMT    50
022772 005137 001244                      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
022776 000240                      79$:  NOP
023000 000004                      1$:  SCOPE          ;LOOP ?
  
```

505
509
523
524

```

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

525
555

```

023002
023002 005737 001274      TST    KYBCTL        ;PERFORMING ONLY SINGLE TESTS ?
023006 001406                      BEQ    2$           ;BR IF NOT
023010 100002                      BPL    1$           ;BR IF JUST ENTERED TEST
023012 000137 003010      JMP    EXEC          ;RETURN & GET NEXT TEST NUMBER
023016 012737 177777 001274  1$:  MOV    #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
023024 112737 000016 001102  2$:  MOV    #16,$STSTM ;TEST NUMBER
023032 012737 023054 001106      MOV    #TEST16,$LPADR ;LOAD LOOP ON TEST ADDRESS
023040 012737 023054 001110      MOV    #TEST16,$LPERR ;LOAD LOOP ON ERROR ADDRESS
023046 012737 000004 001176      MOV    #4,$TIMES    ;DO 4 ITERATIONS
023054 012706 001100  TEST16: MOV    #STACK,SP    ;LOAD THE STACK POINTER
  
```

;SET ATTENTION BITS FOR BOTH PORTS

```

023060 113760 001224 000010      MOV    PORTA,RPCS2(R0) ;SELECT PORT 64$
023066 012760 177777 000014      MOV    #-1,RPER1(R0)  ;FORCE ERRORS
023074 005060 000014                      CLR    RPER1(R0)      ;CLEAR THE ERRORS
023100 013760 001226 000010      MOV    PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
023106 005760 000012  64$:  TST    RPDS1(R0)     ;WAIT FOR DRIVE TO TIMEOUT
023112 001775                      BEQ    64$           ;BR IF DRIVE HASN'T TIMED OUT
023114 012760 177777 000014      MOV    #-1,RPER1(R0)  ;FORCE ERRORS ON PORT 65$
023122 005060 000014                      CLR    RPER1(R0)      ;CLEAR THE ERRORS
023126 113760 001224 000010      MOV    PORTA,RPCS2(R0) ;SELECT PORT '64$' AGAIN
023134 005760 000012  65$:  TST    RPDS1(R0)     ;WAIT FOR DRIVE TO TIMEOUT
023140 001775                      BEQ    65$           ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

023142 113760 001224 000010      MOV    PORTA,RPCS2(R0) ;SELECT PORT A
023150 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

```

023156 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
023162 016037 000012 001126  MOV      RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
023170 012737 000012 001122  MOV      #RPDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
023176 060037 001122          ADD      R0, $BDADR      ;ADD RH11 BASE ADDRESS
023202 012737 100000 001124  MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
023210 013737 001126 001164  MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
023216 042737 077777 001164  BIC      #^CATA, $TMP0   ;SAVE SPECIFIED BITS
023224 023737 001124 001164  CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
023232 001414          BEQ      66$           ;BR IF OK
023234 013737 001126 001174  MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
023242 042737 100000 001174  BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
023250 053737 001174 001124  BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
023256 104010          EMT      10
023260 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
023264 000240          NOP
023266 005737 001244          TST      CKERR          ;WAS ATTN BIT FOR PORT A SET ?
023272 001402          BEQ      .+6          ;BR IF IT WAS
023274 000137 024240          JMP      1$           ;BYPASS REST OF TEST IF NOT
023300 113760 001226 000010  MOVVB   PORTB, RPCS2(R0) ;SELECT PORT B
023306 013737 001226 001234  MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
023314 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
023320 016037 000012 001126  MOV      RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
023326 012737 000012 001122  MOV      #RPDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
023334 060037 001122          ADD      R0, $BDADR      ;ADD RH11 BASE ADDRESS
023340 012737 100000 001124  MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
023346 013737 001126 001164  MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
023354 042737 077777 001164  BIC      #^CATA, $TMP0   ;SAVE SPECIFIED BITS
023362 023737 001124 001164  CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
023370 001414          BEQ      68$           ;BR IF OK
023372 013737 001126 001174  MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
023400 042737 100000 001174  BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
023406 053737 001174 001124  BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
023414 104010          EMT      10
023416 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
023422 000240          NOP
023424 005737 001244          TST      CKERR          ;WAS ATTN BIT FOR PORT B SET ?
023430 001402          BEQ      .+6          ;BR IF IT WAS
023432 000137 024240          JMP      1$           ;BYPASS REST OF TEST IF NOT

;VERIFY THAT THE DRIVE IS IN NEUTRAL

023436 005037 001250          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
023442 012737 000012 001122  MOV      #RPDS1, $BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
023450 060037 001122          ADD      R0, $BDADR      ;ADD THE I/O BASE ADDRESS
023454 012737 111700 001124  MOV      #111700, $GDDAT ;COMPARISON CONSTANT
023462 113760 001224 000010  MOVVB   PORTA, RPCS2(R0) ;SELECT PORT A.
023470 016037 000012 001170  MOV      RPDS1(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
023476 013737 001170 001164  MOV      $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
023504 042737 100100 001164  BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023512 113760 001226 000010  MOVVB   PORTB, RPCS2(R0) ;SELECT PORT B.
023520 016037 000012 001172  MOV      RPDS1(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
023526 013737 001172 001166  MOV      $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
023534 042737 100100 001166  BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023542 023737 001164 001166  CMP      $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023550 001006          BNE      70$           ;BR IF NOT
023552 005737 001164          TST      $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
    
```

```

023556 001045          BNE      72$          ;BR IF NOT
023560 104046          EMT      46
023562 000137 023746   JMP      74$          ;BYPASS THE REST OF THE CHECKS
023566 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023574 013737 001226 001234   MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023602 113760 001226 000010   MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
023610 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
023614 001414          BEQ     71$          ;BR IF ZERO
023616 013737 001224 001234   MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023624 013737 001172 001126   MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
023632 113760 001224 000010   MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
023640 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
023644 001012          BNE     72$          ;BR IF NOT
023646 012737 177777 001250 71$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
023654 012760 000011 000000   MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
023662 012760 000013 000000   MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
023670 104026          EMT      26
023672 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
023700 013737 001224 001234   MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
023706 023737 001124 001170   CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
023714 001401          BEQ     73$          ;BR IF OK FROM PORT A.
023716 104007          EMT      7
023720 013737 001172 001126 73$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
023726 013737 001226 001234   MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
023734 023737 001124 001172   CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
023742 001401          BEQ     74$          ;BR IF OK
023744 104007          EMT      7
023746 000240          NOP
023750 005737 001250          TST     RELERR       ;WAS DRIVE IN NEUTRAL ?
023754 001402          BEQ     +6           ;BR IF IT WAS
023756 000137 024240          JMP     1$           ;BY-PASS RESET OF TEST
;ISSUE THE MASSBUS INIT

023762 012760 000040 000010   MOV     #CLR,RPCS2(R0) ;ISSUE A MASSBUS INIT
;CHECK THE ATTENTION BITS OF BOTH PORTS

023770 113760 001224 000010   MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
023776 013737 001224 001234   MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024004 005037 001244          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
024010 016037 000012 001126   MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
024016 012737 000012 001122   MOV     #RPDS1,$BADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
024024 060037 001122          ADD     R0,$BADR     ;ADD RH11 BASE ADDRESS
024030 012737 100000 001124   MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
024036 013737 001126 001164   MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
024044 042737 077777 001164   BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
024052 023737 001124 001164   CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
024060 001414          BEQ     75$          ;BR IF OK
024062 013737 001126 001174   MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
024070 042737 100000 001174   BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
024076 053737 001174 001124   BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
024104 104051          EMT      51
024106 005137 001244          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
024112 000240          NOP
024114 113760 001226 000010 75$:  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
024122 013737 001226 001234   MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024130 005037 001244          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR

```

```

024134 016037 000012 001126      MOV      RPDS1(R0),SBDDAT ;GET CONTENTS OF RPDS1
024142 012737 000012 001122      MOV      #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
024150 060037 001122      ADD      R0,$BDADR      ;ADD RH11 BASE ADDRESS
024154 012737 100000 001124      MOV      #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
024162 013737 001126 001164      MOV      SBDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
024170 042737 077777 001164      BIC      #^CATA,$TMP0  ;SAVE SPECIFIED BITS
024176 023737 001124 001164      CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
024204 001414      BEQ      77$           ;BR IF OK
024206 013737 001126 001174      MOV      SBDDAT,$TMP4  ;COPY 'BAD DATA'
024214 042737 100000 001174      BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
024222 053737 001174 001124      BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
024230 104051      EMT      51
024232 005137 001244      COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
024236 000240      77$:    NOP
024240 000004      1$:    SCOPE          ;LOOP ?
  
```

556
568
569

```

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

```

024242 005737 001274      TST17:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
024242 001406      BEQ      2$             ;BR IF NOT
024250 100002      BPL      1$            ;BR IF JUST ENTERED TEST
024252 000137 003010      JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
024256 012737 177777 001274  1$:    MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
024264 112737 000017 001102  2$:    MOVVB   #17,$STNM    ;TEST NUMBER
024272 012737 024314 001106      MOV      #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
024300 012737 024314 001110      MOV      #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
024306 012737 007640 001176      MOV      #4000,$TIMES  ;DO 4000. ITERATIONS
570 024314 012706 001100      TEST17: MOV      #STACK,SP ;LOAD THE STACK POINTER
579
  
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

024320 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT #A
024326 005060 000012      CLR      RPDS1(R0)      ;SEIZE THE DRIVE
024332 012760 000011 000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
024340 012760 000013 000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
024346 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT #B
024354 005060 000012      CLR      RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
024360 012760 000011 000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
024366 012760 000013 000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

024374 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
024402 013737 001224 001236      MOV      PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
  
```

```

024410 005760 000000          TST      RPCS1(R0)      ;READ RHCS1
024414 113760 001226 000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
024422 013737 001226 001234  MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024430 013737 001226 001240  MOV     PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
024436 016037 000012 001126  MOV     RPDS1(R0),%BDDAT ;SEE IF DRIVE SEIZED BY PORT A
024444 010037 001122          MOV     R0,%BDADR      ;RH11 BASE ADDRESS
024450 062737 000012 001122  ADD     %RPDS1,%BDADR   ;GENERATE BAD REGISTER ADDRESS
024456 005037 001124          CLR     %GDDAT         ;REGISTER SHOULD BE ZERO
024462 023737 001124 001126  CMP     %GDDAT,%BDDAT  ;IS THE REGISTER ZERO
024470 001403          BEQ    64$            ;BR IF IT IS
024472 104004          EMT    4
024474 000137 025126          JMP    1$            ;BYPASS REST OF THE SUBTEST
024500          64$:
024500 113760 001224 000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
024506 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024514 016037 000012 001126  MOV     RPDS1(R0),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
024522 012737 011700 001124  MOV     %MOL!%PGM!%DPR!%DRY!%VV,%GDDAT ;EXPECTED STATUS
024530 013737 001124 001166  MOV     %GDDAT,%STMP1  ;USE GOOD DATA AS A MASK
024536 005137 001166          COM    %STMP1         ;COMPLEMENT THE EXPECTED STATUS
024542 013737 001126 001164  MOV     %BDDAT,%STMP0  ;SAVE THE ACTUAL STATUS
024550 043737 001166 001164  BIC    %STMP1,%STMP0  ;CLEAR UNWANTED BITS
024556 023737 001124 001164  CMP     %GDDAT,%STMP0  ;ARE THE EXPECTED STATUS BITS SET ?
024564 001401          BEQ    65$            ;BR IF THEY ARE
024566 104005          EMT    5
024570 000240          65$:
                                NOP

                                ;RELEASE THE DRIVE FROM PORT A

024572 113760 001224 000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
024600 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024606 012760 000013 000000  MOV     #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

024614 005037 001250          CLR     RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
024620 012737 000012 001122  MOV     %RPDS1,%BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
024626 060037 001122          ADD    R0,%BDADR      ;ADD THE I/O BASE ADDRESS
024632 012737 011700 001124  MOV     %MOL!%PGM!%DPR!%DRY!%VV,%GDDAT ;COMPARISON CONSTANT
024640 113760 001224 000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
024646 016037 000012 001170  MOV     RPDS1(R0),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
024654 013737 001170 001164  MOV     %STMP2,%STMP0  ;COPY IT INTO 'STMP0'
024662 042737 100100 001164  BIC    %ATA!%VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024670 113760 001226 000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
024676 016037 000012 001172  MOV     RPDS1(R0),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
024704 013737 001172 001166  MOV     %STMP3,%STMP1  ;COPY IT INTO 'STMP1'
024712 042737 100100 001166  BIC    %ATA!%VV,%STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024720 023737 001164 001166  CMP     %STMP0,%STMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
024726 001006          BNE    66$            ;BR IF NOT
024730 005737 001164          TST    %STMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
024734 001045          BNE    68$            ;BR IF NOT
024736 104046          EMT    46
024740 000137 025124          JMP    70$            ;BYPASS THE REST OF THE CHECKS
024744 013737 001170 001126  66$: MOV     %STMP2,%BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
024752 013737 001226 001234  MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024760 113760 001226 000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
024766 005737 001164          TST    %STMP0         ;SEE IF STATUS EQ 0 FROM PORT A.

```



```

024772 001414          BEQ      67$          ;BR IF ZERO
024774 013737 001224 001234  MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025002 013737 001172 001126  MOV     $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
025010 113760 001224 000010  MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
025016 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
025022 001012          BNE     68$          ;BR IF NOT
025024 012737 177777 001250 67$:  MOV     #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
025032 012760 000011 000000  MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
025040 012760 000013 000000  MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
025046 104026          EMT     26
025050 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
025056 013737 001224 001234  MOV     PORTA,PTNBR    ;CHANGE PORT NUMBER
025064 023737 001124 001170  CMP     $GDDAT,$TMP2   ;ALL BITS OK ?
025072 001401          BEQ     69$          ;BR IF OK FROM PORT A.
025074 104007          EMT     7
025076 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
025104 013737 001226 001234  MOV     PORTB,PTNBR    ;CHANGE PORT NUMBER
025112 023737 001124 001172  CMP     $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
025120 001401          BEQ     70$          ;BR IF OK
025122 104007          EMT     7
025124 000240          70$:  NOP
025126 000004          1$:  SCOPE          ;LOOP ?
  
```

591
592

```

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

```

025130          TST20:
025130 005737 001274          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
025134 001406          BEQ     2$          ;BR IF NOT
025136 100002          BPL     1$          ;BR IF JUST ENTERED TEST
025140 000137 003010          JMP     EXEC          ;RETURN & GET NEXT TEST NUMBER
025144 012737 177777 001274 1$:  MOV     #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
025152 112737 000020 001102 2$:  MOVVB  #20,$STNM      ;TEST NUMBER
025160 012737 025202 001106  MOV     #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
025166 012737 025202 001110  MOV     #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
025174 012737 007640 001176  MOV     #4000, $TIMES  ;;DO 4000. ITERATIONS
593 025202 012706 001100  TEST20: MOV    #STACK,SP ;LOAD THE STACK POINTER
594
  
```

593
594

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

025206 113760 001224 000010  MOVVB  PORTA,RPCS2(R0) ;SELECT PORT #A
025214 005060 000012          CLR     RPDS1(R0)     ;SEIZE THE DRIVE
025220 012760 000011 000000  MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
025226 012760 000013 000000  MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
025234 113760 001226 000010  MOVVB  PORTB,RPCS2(R0) ;SELECT PORT #B
025242 005060 000012          CLR     RPDS1(R0)     ;SEIZE THE DRIVE THROUGH PORT 'B'
025246 012760 000011 000000  MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
025254 012760 000013 000000  MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
  
```

:SEIZE THE DRIVE THROUGH PORT B

025262	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	:SELECT PORT B
025270	013737	001226	001236	MOV	PORTB,SEIZPT	:STORE SEIZING PORT'S ADDRESS
025276	005760	000000		TST	RPCS1(R0)	:READ RHCS1
025302	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	:SELECT PORT A
025310	013737	001224	001234	MOV	PORTA,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025316	013737	001224	001240	MOV	PORTA,OPPR	: 'OPPOSITE' PORT ADDRESS
025324	016037	000012	001126	MOV	RPDS1(R0),%BDDAT	:SEE IF DRIVE SEIZED BY PORT B
025332	010037	001122		MOV	R0,%BADDR	:RH11 BASE ADDRESS
025336	062737	000012	001122	ADD	#RPDS1,%BADDR	:GENERATE BAD REGISTER ADDRESS
025344	005037	001124		CLR	%GDDAT	:REGISTER SHOULD BE ZERO
025350	023737	001124	001126	CMP	%GDDAT,%BDDAT	:IS THE REGISTER ZERO
025356	001403			BEQ	64\$:BR IF IT IS
025360	104004			EMT	4	
025362	000137	026014		JMP	1\$:BYPASS REST OF THE SUBTEST
025366						
025366	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	:SELECT PORT B
025374	013737	001226	001234	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025402	016037	000012	001126	MOV	RPDS1(R0),%BDDAT	:SEE IF SEIZING PORT SEES CORRECT STATUS
025410	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,%GDDAT	:EXPECTED STATUS
025416	013737	001124	001166	MOV	%GDDAT,%STMP1	:USE GOOD DATA AS A MASK
025424	005137	001166		COM	%STMP1	:COMPLEMENT THE EXPECTED STATUS
025430	013737	001126	001164	MOV	%BDDAT,%STMP0	:SAVE THE ACTUAL STATUS
025436	043737	001166	001164	BIC	%STMP1,%STMP0	:CLEAR UNWANTED BITS
025444	023737	001124	001164	CMP	%GDDAT,%STMP0	:ARE THE EXPECTED STATUS BITS SET ?
025452	001401			BEQ	65\$:BR IF THEY ARE
025454	104005			EMT	5	
025456	000240			NOP		

:RELEASE THE DRIVE FROM PORT B

025460	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	:SELECT PORT B
025466	013737	001226	001234	MOV	PORTB,PTNBR	:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025474	012760	000013	000000	MOV	#13,RPCS1(R0)	:ISSUE RELEASE THROUGH PORT B

:VERIFY THAT THE DRIVE IS IN NEUTRAL

025502	005037	001250		CLR	RELERR	:CLEAR THE 'RELEASE ERROR' INDICATOR
025506	012737	000012	001122	MOV	#RPDS1,%BADDR	:FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
025514	060037	001122		ADD	R0,%BADDR	:ADD THE I/O BASE ADDRESS
025520	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,%GDDAT	:COMPARISON CONSTANT
025526	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	:SELECT PORT A.
025534	016037	000012	001170	MOV	RPDS1(R0),%STMP2	:GET THE DRIVE STATUS REGISTER FROM PORT A.
025542	013737	001170	001164	MOV	%STMP2,%STMP0	:COPY IT INTO 'STMP0'
025550	042737	100100	001164	BIC	#ATA!VV,%STMP0	:CLEAR PORT DEPENDENT BITS FROM THE COPY
025556	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	:SELECT PORT B.
025564	016037	000012	001172	MOV	RPDS1(R0),%STMP3	:GET THE DRIVE STATUS REGISTER FROM PORT B.
025572	013737	001172	001166	MOV	%STMP3,%STMP1	:COPY IT INTO 'STMP1'
025600	042737	100100	001166	BIC	#ATA!VV,%STMP1	:CLEAR PORT DEPENDENT BITS FROM THE COPY
025606	023737	001164	001166	CMP	%STMP0,%STMP1	:IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
025614	001006			BNE	66\$:BR IF NOT
025616	005737	001164		TST	%STMP0	:REGISTERS ARE THE SAME: ARE THEY ZERO ?
025622	001045			BNE	68\$:BR IF NOT

```

025624 104046          EMT      46
025626 000137 026012   JMP      70$      ;BYPASS THE REST OF THE CHECKS
025632 013737 001170 001126 66$:   MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
025640 013737 001226 001234   MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025646 113760 001226 000010   MOVVB  PORTB,RPCS2(R0) ;SELECT PORT B.
025654 005737 001164          TST     $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
025660 001414          BEQ     67$      ;BR IF ZERO
025662 013737 001224 001234   MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025670 013737 001172 001126   MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
025676 113760 001224 000010   MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
025704 005737 001166          TST     $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
025710 001012          BNE     68$      ;BR IF NOT
025712 012737 177777 001250 67$:   MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
025720 012760 000011 000000   MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
025726 012760 000013 000000   MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
025734 104026          EMT      26
025736 013737 001170 001126 68$:   MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
025744 013737 001224 001234   MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
025752 023737 001124 001170   CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
025760 001401          BEQ     69$      ;BR IF OK FROM PORT A.
025762 104007          EMT      7
025764 013737 001172 001126 69$:   MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
025772 013737 001226 001234   MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
026000 023737 001124 001172   CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
026006 001401          BEQ     70$      ;BR IF OK
026010 104007          EMT      7
026012 000240          70$:   NOP
026014 000004          1$:   SCOPE      ;LOOP ?
    
```

595
614
615

```

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

```

026016          TST21:
026016 005737 001274   TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
026022 001406          BEQ     2$      ;BR IF NOT
026024 100002          BPL     1$      ;BR IF JUST ENTERED TEST
026026 000137 003010   JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
026032 012737 177777 001274 1$:   MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
026040 112737 000021 001102 2$:   MOVVB  #21,$STNM   ;TEST NUMBER
026046 012737 026070 001106   MOV     #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS
    
```

```

026054 012737 026070 001110      MOV    #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS
026062 012737 007640 001176      MOV    #4000.,$TIMES  ;;DO 4000. ITERATIONS
616 026070 012706 001100      TEST21: MOV   #STACK,SP ;LOAD THE STACK POINTER
644
;CLEAR ATTENTION BITS FOR BOTH PORTS

026074 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
026102 005060 000012                CLR    RPDS1(R0) ;SEIZE THE DRIVE
026106 012760 000011 000000      MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
026114 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
026122 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT #B
026130 005060 000012                CLR    RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
026134 012760 000011 000000      MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
026142 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

026150 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
026156 013737 001226 001236      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
026164 005060 000012                CLR    RPDS1(R0) ;WRITE RPDS1
026170 013737 001224 001240      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
026176 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
026204 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET PORT REQUEST

026212 005060 000012                CLR    RPDS1(R0) ;SET PORT REQUEST FOR PORT A

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

;RELEASE THE DRIVE FROM PORT B

026216 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
026224 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026232 012760 000013 000000      MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B

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

026240 005037 001250                CLR    RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
026244 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
026252 012737 000012 001122      MOV    #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
026260 060037 001122                ADD    R0,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
026264 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
026272 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026300 016037 000012 001164      MOV    RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
026306 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
026314 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026322 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT B
026330 001404                BEQ    66$ ;BR IF STATUS FROM PORT B ZERO
026332 005737 001164                TST    $TMP0 ;IS STATUS FROM PORT A ZERO?
026336 001401                BEQ    66$ ;BR IF ZERO
026340 104031                EMT    31
026342 013737 001164 001126      66$: MOV    $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
026350 013737 001224 001234      MOV    PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
026356 023737 001124 001126      CMP    $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
  
```

```

026364 001401      BEQ      67$      ;BR IF OK
026366 104027      EMT      27
026370 000240      NOP
026372 113760 001226 000010 67$: MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
026400 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026406 005037 001244      CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
026412 016037 000012 001126      MOV    RPDS1(R0),%SBDAT ;GET CONTENTS OF RPDS1
026420 012737 000012 001122      MOV    %RPDS1,%SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
026426 060037 001122      ADD    R0,%SBDADR  ;ADD RH11 BASE ADDRESS
026432 005037 001124      CLR    %GDDAT     ;WHAT REGISTER SHOULD BE
026436 013737 001126 001164      MOV    %SBDAT,%STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
026444 042737 077777 001164      BIC    %^CATA,%STMP0 ;SAVE SPECIFIED BITS
026452 023737 001124 001164      CMP    %GDDAT,%STMP0 ;COMPARE THE BITS
026460 001414      BEQ    68$      ;BR IF OK
026462 013737 001126 001174      MOV    %SBDAT,%STMP4 ;COPY 'BAD DATA'
026470 042737 100000 001174      BIC    %ATA,%STMP4  ;CLEAR THE MASKED BITS
026476 053737 001174 001124      BIS    %STMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
026504 104016      EMT      16
026506 005137 001244      COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
026512 000240      NOP
026514 113760 001224 000010 68$: MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
026522 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026530 005037 001244      CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
026534 016037 000012 001126      MOV    RPDS1(R0),%SBDAT ;GET CONTENTS OF RPDS1
026542 012737 000012 001122      MOV    %RPDS1,%SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
026550 060037 001122      ADD    R0,%SBDADR  ;ADD RH11 BASE ADDRESS
026554 012737 100000 001124      MOV    %ATA,%GDDAT  ;WHAT REGISTER SHOULD BE
026562 013737 001126 001164      MOV    %SBDAT,%STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
026570 042737 077777 001164      BIC    %^CATA,%STMP0 ;SAVE SPECIFIED BITS
026576 023737 001124 001164      CMP    %GDDAT,%STMP0 ;COMPARE THE BITS
026604 001414      BEQ    70$      ;BR IF OK
026606 013737 001126 001174      MOV    %SBDAT,%STMP4 ;COPY 'BAD DATA'
026614 042737 100000 001174      BIC    %ATA,%STMP4  ;CLEAR THE MASKED BITS
026622 053737 001174 001124      BIS    %STMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
026630 104016      EMT      16
026632 005137 001244      COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
026636 000240      NOP
70$:

;RELEASE THE DRIVE FROM PORT A

026640 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
026646 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026654 012760 000013 000000      MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

026662 005037 001250      CLR    RELERR     ;CLEAR THE 'RELEASE ERROR' INDICATOR
026666 012737 000012 001122      MOV    %RPDS1,%SBDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
026674 060037 001122      ADD    R0,%SBDADR  ;ADD THE I/O BASE ADDRESS
026700 012737 011700 001124      MOV    %MOL!%PGM!%DPR!%DRY!%VV,%GDDAT ;COMPARISON CONSTANT
026706 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
026714 016037 000012 001170      MOV    RPDS1(R0),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
026722 013737 001170 001164      MOV    %STMP2,%STMP0 ;COPY IT INTO 'STMP0'
026730 042737 100100 001164      BIC    %ATA!%VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026736 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
026744 016037 000012 001172      MOV    RPDS1(R0),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

```

026752 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
026760 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026766 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
026774 001006                        BNE      72$             ;BR IF NOT
026776 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
027002 001045                        BNE      74$             ;BR IF NOT
027004 104046                        EMT      46
027006 000137 027172                JMP      76$             ;BYPASS THE REST OF THE CHECKS
027012 013737 001170 001126 72$:   MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
027020 013737 001226 001234        MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027026 113760 001226 000010        MOVB     PORTB,RPCS2(R0) ;SELECT PORT B.
027034 005737 001164                TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
027040 001414                        BEQ      73$             ;BR IF ZERO
027042 013737 001224 001234        MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027050 013737 001172 001126        MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
027056 113760 001224 000010        MOVB     PORTA,RPCS2(R0) ;SELECT PORT A.
027064 005737 001166                TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
027070 001012                        BNE      74$             ;BR IF NOT
027072 012737 177777 001250 73$:   MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
027100 012760 000011 000000        MOV      #11,RPCS1(R0)  ;CLEAR THE DRIVE
027106 012760 000013 000000        MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
027114 104026                        EMT      26
027116 013737 001170 001126 74$:   MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
027124 013737 001224 001234        MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
027132 023737 001124 001170        CMP      $GDDAT,$TMP2   ;ALL BITS OK ?
027140 001401                        BEQ      75$             ;BR IF OK FROM PORT A.
027142 104007                        EMT      7
027144 013737 001172 001126 75$:   MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
027152 013737 001226 001234        MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
027160 023737 001124 001172        CMP      $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
027166 001401                        BEQ      76$             ;BR IF OK
027170 104007                        EMT      7
027172 000240                        NOP
027174 000004 76$:                 NOP
1$:                                     SCOPE ;LOOP ?
  
```

16
17

```

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

```

027176
027176 005737 001274
027202 001406
027204 100002
027206 000137 003010
027212 012737 177777 001274
027220 112737 000022 001102
027226 012737 027250 001106
027234 012737 027250 001110
027242 012737 007640 001176
18 027250 012706 001100
19

```

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$:   MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
2$:   MOVB    #22,$TSTNM  ;TEST NUMBER
      MOV      #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #4000,$TIMES ;DO 4000. ITERATIONS
TEST22: MOV     #STACK,SP  ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

027254 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT #A
027262 005060 000012          CLR   RPDS1(R0)       ;SEIZE THE DRIVE
027266 012760 000011 000000  MOV   #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
027274 012760 000013 000000  MOV   #13,RPCS1(R0)   ;RELEASE THE DRIVE
027302 113760 001226 000010  MOVB  PORTB,RPCS2(R0) ;SELECT PORT #B
027310 005060 000012          CLR   RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
027314 012760 000011 000000  MOV   #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
027322 012760 000013 000000  MOV   #13,RPCS1(R0)   ;RELEASE THE DRIVE

```

;SEIZE THE DRIVE THROUGH PORT A

```

027330 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT A
027336 013737 001224 001236  MOV   PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
027344 005060 000012          CLR   RPDS1(R0)       ;WRITE RPDS1
027350 013737 001226 001240  MOV   PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
027356 113760 001226 000010  MOVB  PORTB,RPCS2(R0) ;SELECT PORT B
027364 013737 001226 001234  MOV   PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

;SET PORT REQUEST

```

027372 005060 000012          CLR   RPDS1(R0)       ;SET PORT REQUEST FOR PORT B

```

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

;RELEASE THE DRIVE FROM PORT A

027376	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	;SELECT PORT A
027404	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027412	012760	000013	000000	MOV	#13,RPCS1(R0)	;ISSUE RELEASE THROUGH PORT A

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

027420	005037	001250		CLR	RELERR	;CLEAR 'RELEASE ERROR' INDICATOR
027424	012737	111700	001124	MOV	#ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT	;COMPARISON CONSTANT
027432	012737	000012	001122	MOV	#RPDS1,\$BDADR	;REGISTER ADDRESS INCREMENT
027440	060037	001122		ADD	R0,\$BDADR	;REGISTER BASE ADDRESS FOR TYPEOUT
027444	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	;SELECT PORT B
027452	013737	001226	001234	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027460	016037	000012	001164	MOV	RPDS1(R0),\$TMP0	;READ STATUS REGISTER FROM PORT B
027466	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	;SELECT PORT A
027474	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027502	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;DRIVE STATUS FROM PORT A
027510	001404			BEQ	66\$;BR IF STATUS FROM PORT A ZERO
027512	005737	001164		TST	\$TMP0	;IS STATUS FROM PORT B ZERO ?
027516	001401			BEQ	66\$;BR IF ZERO
027520	104031			EMT	31	
027522	013737	001164	001126	66\$: MOV	\$TMP0,\$BDDAT	;CHECK STATUS FROM PORT B
027530	013737	001226	001234	MOV	PORTB,PTNBR	;CHANGE PORT ADDRESS FOR TYPEOUT
027536	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;COMPARE WITH CONSTANT
027544	001401			BEQ	67\$;BR IF OK
027546	104027			EMT	27	
027550	000240			67\$: NOP		
027552	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	;SELECT PORT A
027560	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027566	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
027572	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
027600	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
027606	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
027612	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
027616	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
027624	042737	077777	001164	BIC	#^CATA,\$TMP0	;SAVE SPECIFIED BITS
027632	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;COMPARE THE BITS
027640	001414			BEQ	68\$;BR IF OK
027642	013737	001126	001174	MOV	\$BDDAT,\$TMP4	;COPY 'BAD DATA'
027650	042737	100000	001174	BIC	#ATA,\$TMP4	;CLEAR THE MASKED BITS
027656	053737	001174	001124	BIS	\$TMP4,\$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
027664	104016			EMT	16	
027666	005137	001244		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
027672	000240			68\$: NOP		
027674	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	;SELECT PORT B
027702	013737	001226	001234	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027710	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
027714	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
027722	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
027730	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
027734	012737	100000	001124	MOV	#ATA,\$GDDAT	;WHAT REGISTER SHOULD BE
027742	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
027750	042737	077777	001164	BIC	#^CATA,\$TMP0	;SAVE SPECIFIED BITS


```

027756 023737 001124 001164      CMP      $GDDAT,$STMP0      ;COMPARE THE BITS
027764 001414                      BEQ      70$                ;BR IF OK
027766 013737 001126 001174      MOV      $BDDAT,$STMP4     ;COPY 'BAD DATA'
027774 042737 100000 001174      BIC      #ATA,$STMP4       ;CLEAR THE MASKED BITS
030002 053737 001174 001124      BIS      $STMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
030010 104016                      EMT      16
030012 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
030016 000240                      70$:  NOP

                                ;RELEASE THE DRIVE FROM PORT B

030020 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B
030026 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030034 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

030042 005037 001250                      CLR      RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
030046 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
030054 060037 001122                      ADD      R0,$BDADR        ;ADD THE I/O BASE ADDRESS
030060 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
030066 113760 001224 000010      MOVB     PORTA,RPCS2(R0)   ;SELECT PORT A.
030074 016037 000012 001170      MOV      RPDS1(R0),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
030102 013737 001170 001164      MOV      $STMP2,$STMP0    ;COPY IT INTO '$STMP0'
030110 042737 100100 001164      BIC      #ATA!VV,$STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030116 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B.
030124 016037 000012 001172      MOV      RPDS1(R0),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
030132 013737 001172 001166      MOV      $STMP3,$STMP1   ;COPY IT INTO '$STMP1'
030140 042737 100100 001166      BIC      #ATA!VV,$STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030146 023737 001164 001166      CMP      $STMP0,$STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
030154 001006                      BNE      72$              ;BR IF NOT
030156 005737 001164                      TST      $STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
030162 001045                      BNE      74$              ;BR IF NOT
030164 104046                      EMT      46
030166 000137 030352                      JMP      76$              ;BYPASS THE REST OF THE CHECKS
030172 013737 001170 001126      72$:  MOV      $STMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
030200 013737 001226 001234      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030206 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B.
030214 005737 001164                      TST      $STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
030220 001414                      BEQ      73$              ;BR IF ZERO
030222 013737 001224 001234      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030230 013737 001172 001126      MOV      $STMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
030236 113760 001224 000010      MOVB     PORTA,RPCS2(R0)   ;SELECT PORT A.
030244 005737 001166                      TST      $STMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
030250 001012                      BNE      74$              ;BR IF NOT
030252 012737 177777 001250      73$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
030260 012760 000011 000000      MOV      #11,RPCS1(R0)    ;CLEAR THE DRIVE
030266 012760 000013 000000      MOV      #13,RPCS1(R0)    ;RELEASE THE DRIVE
030274 104026                      EMT      26
030276 013737 001170 001126      74$:  MOV      $STMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
030304 013737 001224 001234      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
030312 023737 001124 001170      CMP      $GDDAT,$STMP2   ;ALL BITS OK ?
030320 001401                      BEQ      75$              ;BR IF OK FROM PORT A.
030322 104007                      EMT      7
030324 013737 001172 001126      75$:  MOV      $STMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
030332 013737 001226 001234      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER

```

```

030340 023737 001124 001172      CMP      $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
030346 001401                      BEQ      76$              ;BR IF OK
030350 104007                      EMT      7
030352 000240                      76$:    NOP
030354 000004                      1$:    SCOPE              ;LOOP ?
    
```

20
36
37

```

:*****
:*TEST 23      TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
:*
:*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
:*REQUEST'.
:*
:*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1.  VERIFY THAT
:*      THE DRIVE HAS BEEN SEIZED.
:*
:*  B.  READ THE CONTROL REGISTER FROM PORT 'A'.  VERIFY THAT 'DVA' IS NOT
:*      SET.
:*
:*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
:*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*
:*****
    
```

```

030356 005737 001274      TST      KYBCTL           ;PERFORMING ONLY SINGLE TESTS ?
030356 001406                      BEQ      2$              ;BR IF NOT
030364 100002                      BPL      1$              ;BR IF JUST ENTERED TEST
030366 000137 003010      JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
030372 012737 177777 001274 1$:    MOV      #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
030400 112737 000023 001102 2$:    MOV      #23,$TSTNM      ;TEST NUMBER
030406 012737 030430 001106      MOV      #TEST23,$LPADR  ;LOAD LOOP ON TEST ADDRESS
030414 012737 030430 001110      MOV      #TEST23,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
030422 012737 007640 001176      MOV      #4000,$TIMES   ;;DO 4000. ITERATIONS
38 030430 012706 001100  TEST23: MOV      #STACK,$SP    ;LOAD THE STACK POINTER
54
    
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

030434 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT #A
030442 005060 000012                      CLR      RPDS1(R0)       ;SEIZE THE DRIVE
030446 012760 000011 000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
030454 012760 000013 000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
030462 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT #B
030470 005060 000012                      CLR      RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
030474 012760 000011 000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
030502 012760 000013 000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
    
```

;SEIZE THE DRIVE THROUGH PORT B

```

030510 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT B
030516 013737 001226 001236      MOV      PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
030524 005760 000000                      TST      RPCS1(R0)       ;READ RHCS1
030530 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A
030536 013737 001224 001234      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030544 013737 001224 001240      MOV      PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
030552 016037 000012 001126      MOV      RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
030560 010037 001122                      MOV      R0,$BDADR      ;RH11 BASE ADDRESS
    
```

```

030564 062737 000012 001122      ADD    #RPDS1,$BDADR    ;GENERATE BAD REGISTER ADDRESS
030572 005037 001124      CLR    $GDDAT          ;REGISTER SHOULD BE ZERO
030576 023737 001124 001126      CMP    $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
030604 001403      BEQ    64$            ;BR IF IT IS
030606 104004      EMT    4
030610 000137 031364      JMP    1$             ;BYPASS REST OF THE SUBTEST
030614      64$:
030614 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
030622 013737 001226 001234      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030630 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
030636 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
030644 013737 001124 001166      MOV    $GDDAT,$STMP1  ;USE GOOD DATA AS A MASK
030652 005137 001166      COM    $STMP1          ;COMPLEMENT THE EXPECTED STATUS
030656 013737 001126 001164      MOV    $BDDAT,$STMP0  ;SAVE THE ACTUAL STATUS
030664 043737 001166 001164      BIC    $STMP1,$STMP0  ;CLEAR UNWANTED BITS
030672 023737 001124 001164      CMP    $GDDAT,$STMP0  ;ARE THE EXPECTED STATUS BITS SET ?
030700 001401      BEQ    65$            ;BR IF THEY ARE
030702 104005      EMT    5
030704 000240      65$:
030706 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
030714 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;READ RPCS1 THROUGH PORT A - TRY TO SET PORT REQUEST

030722 005037 001244      CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
030726 016037 000000 001126      MOV    RPCS1(R0),$BDDAT ;GET CONTENTS OF RPCS1
030734 012737 000000 001122      MOV    #RPCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
030742 060037 001122      ADD    R0,$BDADR       ;ADD RH11 BASE ADDRESS
030746 005037 001124      CLR    $GDDAT          ;WHAT REGISTER SHOULD BE
030752 013737 001126 001164      MOV    $BDDAT,$STMP0  ;MOVE REGISTER CONTENTS TO '$STMP0'
030760 042737 173700 001164      BIC    #^C4077,$STMP0 ;SAVE SPECIFIED BITS
030766 023737 001124 001164      CMP    $GDDAT,$STMP0  ;COMPARE THE BITS
030774 001414      BEQ    66$            ;BR IF OK
030776 013737 001126 001174      MOV    $BDDAT,$STMP4  ;COPY 'BAD DATA'
031004 042737 004077 001174      BIC    #4077,$STMP4   ;CLEAR THE MASKED BITS
031012 053737 001174 001124      BIS    $STMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
031020 104010      EMT    10
031022 005137 001244      COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
031026 000240      66$:
                                NOP

;DRIVE SHOULD RETURN TO NEUTRAL

                                ;RELEASE THE DRIVE FROM PORT B

031030 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
031036 013737 001226 001234      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031044 012760 000013 000000      MOV    #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

031052 005037 001250      CLR    RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
031056 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
031064 060037 001122      ADD    R0,$BDADR       ;ADD THE I/O BASE ADDRESS
031070 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
031076 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
031104 016037 000012 001170      MOV    RPDS1(R0),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
    
```

```

031112 013737 001170 001164      MOV      $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
031120 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031126 113760 001226 000010      MOVVB   PORTB,RPCS2(R0)  ;SELECT PORT B.
031134 016037 000012 001172      MOV      RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
031142 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
031150 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031156 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031164 001006      BNE      68$             ;BR IF NOT
031166 005737 001164      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
031172 001045      BNE      70$             ;BR IF NOT
031174 104046      EMT      46
031176 000137 031362      JMP      72$             ;BYPASS THE REST OF THE CHECKS
031202 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031210 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031216 113760 001226 000010      MOVVB   PORTB,RPCS2(R0)  ;SELECT PORT B.
031224 005737 001164      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
031230 001414      BEQ      69$             ;BR IF ZERO
031232 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031240 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
031246 113760 001224 000010      MOVVB   PORTA,RPCS2(R0)  ;SELECT PORT A.
031254 005737 001166      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
031260 001012      BNE      70$             ;BR IF NOT
031262 012737 177777 001250 69$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
031270 012760 000011 000000      MOV      #11,RPCS1(R0)   ;CLEAR THE DRIVE
031276 012760 000013 000000      MOV      #13,RPCS1(R0)   ;RELEASE THE DRIVE
031304 104026      EMT      26
031306 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
031314 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
031322 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
031330 001401      BEQ      71$             ;BR IF OK FROM PORT A.
031332 104007      EMT      7
031334 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
031342 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
031350 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
031356 001401      BEQ      72$             ;BR IF OK
031360 104007      EMT      7
031362 000240 72$:  NOP
031364 000004 1$:   SCOPE                ;LOOP ?
    
```

70
71

```

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

031366
031366 005737 001274

```

TST24:      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
    
```

```

031372 001406          BEQ      2$          ;BR IF NOT
031374 100002          BPL      1$          ;BR IF JUST ENTERED TEST
031376 000137 003010  JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
031402 012737 177777 001274 1$:    MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
031410 112737 000024 001102 2$:    MOV      #24,$TSTNM ;TEST NUMBER
031416 012737 031440 001106      MOV      #TEST24,$LPADR ;LOAD LOOP ON TEST ADDRESS
031424 012737 031440 001110      MOV      #TEST24,$LPERR ;LOAD LOOP ON ERROR ADDRESS
031432 012737 007640 001176      MOV      #4000,$TIMES  ;DO 4000. ITERATIONS
72 031440 012706 001100  TEST24: MOV     #STACK,SP ;LOAD THE STACK POINTER
73                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS

031444 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT #A
031452 005060 000012          CLR      RPDS1(R0)      ;SEIZE THE DRIVE
031456 012760 000011 000000  MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
031464 012760 000013 000000  MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
031472 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT #B
031500 005060 000012          CLR      RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
031504 012760 000011 000000  MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
031512 012760 000013 000000  MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE

                                     ;SEIZE THE DRIVE THROUGH PORT A

031520 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A
031526 013737 001224 001236  MOV      PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
031534 005760 000000          TST      RPCS1(R0)      ;READ RHCS1
031540 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B
031546 013737 001226 001234  MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031554 013737 001226 001240  MOV      PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
031562 016037 000012 001126  MOV      RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
031570 010037 001122          MOV      R0,$BADDR     ;RH11 BASE ADDRESS
031574 062737 000012 001122  ADD      #RPDS1,$BADDR  ;GENERATE BAD REGISTER ADDRESS
031602 005037 001124          CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
031606 023737 001124 001126  CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
031614 001403          BEQ      64$          ;BR IF IT IS
031616 104004          EMT      4
031620 000137 032374          JMP      1$           ;BYPASS REST OF THE SUBTEST
031624                                     64$:
031624 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A
031632 013737 001224 001234  MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031640 016037 000012 001126  MOV      RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
031646 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
031654 013737 001124 001166  MOV      $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
031662 005137 001166          COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
031666 013737 001126 001164  MOV      $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
031674 043737 001166 001164  BIC      $TMP1,$TMP0   ;CLEAR UNWANTED BITS
031702 023737 001124 001164  CMP      $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
031710 001401          BEQ      65$          ;BR IF THEY ARE
031712 104005          EMT      5
031714 000240          NOP
031716 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B
031724 013737 001226 001234  MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                     ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST

031732 005037 001244          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
    
```

```

031736 016037 000000 001126      MOV      RPCS1(R0), $BDDAT      ;GET CONTENTS OF RPCS1
031744 012737 000000 001122      MOV      #RPCS1, $BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
031752 060037 001122                ADD      R0, $BDADR          ;ADD RH11 BASE ADDRESS
031756 005037 001124                CLR      $GDDAT             ;WHAT REGISTER SHOULD BE
031762 013737 001126 001164      MOV      $BDDAT, $TMP0       ;MOVE REGISTER CONTENTS TO '$TMP0'
031770 042737 173700 001164      BIC      #^C4077, $TMP0      ;SAVE SPECIFIED BITS
031776 023737 001124 001164      CMP      $GDDAT, $TMP0      ;COMPARE THE BITS
032004 001414                BEQ      66$                 ;BR IF OK
032006 013737 001126 001174      MOV      $BDDAT, $TMP4       ;COPY 'BAD DATA'
032014 042737 004077 001174      BIC      #4077, $TMP4        ;CLEAR THE MASKED BITS
032022 053737 001174 001124      BIS      $TMP4, $GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
032030 104010                EMT      10
032032 005137 001244                COM      CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
032036 000240                66$:      NOP

;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT A

032040 113760 001224 000010      MOV      PORTA, RPCS2(R0)    ;SELECT PORT A
032046 013737 001224 001234      MOV      PORTA, PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032054 012760 000013 000000      MOV      #13, RPCS1(R0)     ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

032062 005037 001250                CLR      RELERR             ;CLEAR THE 'RELEASE ERROR' INDICATOR
032066 012737 000012 001122      MOV      #RPDS1, $BDADR     ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
032074 060037 001122                ADD      R0, $BDADR         ;ADD THE I/O BASE ADDRESS
032100 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
032106 113760 001224 000010      MOV      PORTA, RPCS2(R0)   ;SELECT PORT A.
032114 016037 000012 001170      MOV      RPDS1(R0), $TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
032122 013737 001170 001164      MOV      $TMP2, $TMP0       ;COPY IT INTO '$TMP0'
032130 042737 100100 001164      BIC      #ATA!VV, $TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032136 113760 001226 000010      MOV      PORTB, RPCS2(R0)   ;SELECT PORT B.
032144 016037 000012 001172      MOV      RPDS1(R0), $TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
032152 013737 001172 001166      MOV      $TMP3, $TMP1       ;COPY IT INTO '$TMP1'
032160 042737 100100 001166      BIC      #ATA!VV, $TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032166 023737 001164 001166      CMP      $TMP0, $TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
032174 001006                BNE      68$                 ;BR IF NOT
032176 005737 001164                TST      $TMP0               ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
032202 001045                BNE      70$                 ;BR IF NOT
032204 104046                EMT      46
032206 000137 032372                JMP      72$                 ;BYPASS THE REST OF THE CHECKS
032212 013737 001170 001126 68$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
032220 013737 001226 001234      MOV      PORTB, PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
032226 113760 001226 000010      MOV      PORTB, RPCS2(R0)   ;SELECT PORT B.
032234 005737 001164                TST      $TMP0               ;SEE IF STATUS EQ 0 FROM PORT A.
032240 001414                BEQ      69$                 ;BR IF ZERO
032242 013737 001224 001234      MOV      PORTA, PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
032250 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
032256 113760 001224 000010      MOV      PORTA, RPCS2(R0)   ;SELECT PORT A.
032264 005737 001166                TST      $TMP1               ;SEE IF STATUS EQ ZERO FROM PORT B.
032270 001012                BNE      70$                 ;BR IF NOT
032272 012737 177777 001250 69$:  MOV      #-1, RELERR        ;SET 'RELEASE ERROR' INDICATOR
032300 012760 000011 000000      MOV      #11, RPCS1(R0)     ;CLEAR THE DRIVE
032306 012760 000013 000000      MOV      #13, RPCS1(R0)     ;RELEASE THE DRIVE

```

```

032314 104026          EMT      26
032316 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
032324 013737 001224 001234      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
032332 023737 001124 001170      CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
032340 001401          BEQ     71$ ;BR IF OK FROM PORT A.
032342 104007          EMT      7
032344 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
032352 013737 001226 001234      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
032360 023737 001124 001172      CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
032366 001401          BEQ     72$ ;BR IF OK
032370 104007          EMT      7
032372 000240          72$:  NOP
032374 000004          1$:   SCOPE ;LOOP ?

```

74
93
94

```

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

```

```

032376          TST25:
032376 005737 001274      TST     KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
032402 001406          BEQ     2$ ;BR IF NOT
032404 100002          BPL     1$ ;BR IF JUST ENTERED TEST
032406 000137 003010      JMP     EXEC ;RETURN & GET NEXT TEST NUMBER
032412 012737 177777 001274 1$:  MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
032420 112737 000025 001102 2$:  MOV     #25,$TSTNM ;TEST NUMBER
032426 012737 032450 001106      MOV     #TEST25,$LPADR ;LOAD LOOP ON TEST ADDRESS
032434 012737 032450 001110      MOV     #TEST25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
032442 012737 007640 001176      MOV     #4000,$TIMES ;DO 4000. ITERATIONS
95 032450 012706 001100      TEST25: MOV     #STACK,SP ;LOAD THE STACK POINTER
117

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

032454 113760 001224 000010      MOV     PORTA,RPCS2(R0) ;SELECT PORT #A
032462 005060 000012          CLR     RPDS1(R0) ;SEIZE THE DRIVE
032466 012760 000011 000000      MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
032474 012760 000013 000000      MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
032502 113760 001226 000010      MOV     PORTB,RPCS2(R0) ;SELECT PORT #B
032510 005060 000012          CLR     RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
032514 012760 000011 000000      MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
032522 012760 000013 000000      MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE

```

;SEIZE THE DRIVE THROUGH PORT B

032530	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	;SELECT PORT B
032536	013737	001226	001236	MOV	PORTB,SEIZPT	;STORE SEIZING PORT'S ADDRESS
032544	005060	000012		CLR	RPDS1(R0)	;WRITE RPDS1
032550	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	;SELECT PORT A
032556	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032564	013737	001224	0C1240	MOV	PORTA,OPPR	; 'OPPOSITE' PORT ADDRESS
032572	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;SEE IF DRIVE SEIZED BY PORT B
032600	010037	001122		MOV	R0,\$BDADR	;RH11 BASE ADDRESS
032604	062737	000012	001122	ADD	#RPDS1,\$BDADR	;GENERATE BAD REGISTER ADDRESS
032612	005037	001124		CLR	\$GDDAT	;REGISTER SHOULD BE ZERO
032616	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;IS THE REGISTER ZERO
032624	001403			BEQ	64\$;BR IF IT IS
032626	104004			EMT	4	
032630	000137	033602		JMP	1\$;BYPASS REST OF THE SUBTEST
032634						
032634	113760	001226	000010	MOV B	PORTB,RPCS2(R0)	;SELECT PORT B
032642	013737	001226	001234	MOV	PORTB,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032650	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;SEE IF SEIZING PORT SEES CORRECT STATUS
032656	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV,\$GDDAT	;EXPECTED STATUS
032664	013737	001124	001166	MOV	\$GDDAT,\$TMP1	;USE GOOD DATA AS A MASK
032672	005137	001166		COM	\$TMP1	;COMPLEMENT THE EXPECTED STATUS
032676	013737	001126	001164	MOV	\$BDDAT,\$TMP0	;SAVE THE ACTUAL STATUS
032704	043737	001166	001164	BIC	\$TMP1,\$TMP0	;CLEAR UNWANTED BITS
032712	023737	001124	001164	CMP	\$GDDAT,\$TMP0	;ARE THE EXPECTED STATUS BITS SET ?
032720	001401			BEQ	65\$;BR IF THEY ARE
032722	104005			EMT	5	
032724	000240			NOP		

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

032726	113760	001224	000010	MOV B	PORTA,RPCS2(R0)	;SELECT PORT A
032734	013737	001224	001234	MOV	PORTA,PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032742	012760	000013	000000	MOV	#13,RPCS1(R0)	;ISSUE A RELEASE COMMAND THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

032750	005037	001244		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
032754	016037	000012	001126	MOV	RPDS1(R0),\$BDDAT	;GET CONTENTS OF RPDS1
032762	012737	000012	001122	MOV	#RPDS1,\$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
032770	060037	001122		ADD	R0,\$BDADR	;ADD RH11 BASE ADDRESS
032774	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
033000	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;IS THE REGISTER OK ?
033006	001403			BEQ	66\$;BR IF OK
033010	104010			EMT	10	
033012	005137	001244		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
033016	016037	000000	001126	MOV	RPCS1(R0),\$BDDAT	;GET THE CONTENTS OF RHCS1
033024	012737	000000	001122	MOV	#RPCS1,\$BDADR	;FORM ADDRESS OF REGISTER
033032	060037	001122		ADD	R0,\$BDADR	;ADDRESS BASE
033036	032737	020000	001126	BIT	#MCPE,\$BDDAT	;IS 'MCPE' SET ?
033044	001404			BEQ	67\$;BR IF NOT
033046	104011			EMT	11	
033050	012760	040000	000000	MOV	#TRE,RPCS1(R0)	;CLEAR 'MCPE'
033056	000240			NOP		
033060	005737	001244		TST	CKERR	;WAS RPDS1 NON ZERO ?

033064 001402 BEQ +6 ;CONTENTS OF RPDS1 SEEN BY PORT A
 033066 000137 033602 JMP 1\$;DRIVE IN NEUTRAL, BYPASS REST OF TEST

;RELEASE THE DRIVE FROM PORT B

033072 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
 033100 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 033106 012760 000013 000000 MOV #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B

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

033114 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
 033120 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
 033126 012737 000012 001122 MOV #RPDS1,\$BDADR ;REGISTER ADDRESS INCREMENT
 033134 060037 001122 ADD R0,\$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
 033140 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
 033146 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 033154 016037 000012 001164 MOV RPDS1(R0),\$TMP0 ;READ STATUS REGISTER FROM PORT A
 033162 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
 033170 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 033176 016037 000012 001126 MOV RPDS1(R0),\$BDDAT ;DRIVE STATUS FROM PORT B
 033204 001404 BEQ 68\$;BR IF STATUS FROM PORT B ZERO
 033206 005737 001164 TST \$TMP0 ;IS STATUS FROM PORT A ZERO ?
 033212 001401 BEQ 68\$;BR IF ZERO
 033214 104031 EMT 31
 033216 013737 001164 001126 68\$: MOV \$TMP0,\$BDDAT ;CHECK STATUS FROM PORT A
 033224 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
 033232 023737 001124 001126 CMP \$GDDAT,\$BDDAT ;COMPARE WITH CONSTANT
 033240 001401 BEQ 69\$;BR IF OK
 033242 104027 EMT 27
 033244 000240 69\$: NOP

;RELEASE THE DRIVE FROM PORT A

033246 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
 033254 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 033262 012760 000013 000000 MOV #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

033270 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
 033274 012737 000012 001122 MOV #RPDS1,\$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 033302 060037 001122 ADD R0,\$BDADR ;ADD THE I/O BASE ADDRESS
 033306 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
 033314 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
 033322 016037 000012 001170 MOV RPDS1(R0),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 033330 013737 001170 001164 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
 033336 042737 100100 001164 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 033344 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
 033352 016037 000012 001172 MOV RPDS1(R0),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 033360 013737 001172 001166 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
 033366 042737 100100 001166 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 033374 023737 001164 001166 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 033402 001006 BNE 70\$;BR IF NOT
 033404 005737 001164 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 033410 001045 BNE 72\$;BR IF NOT

```

033412 104046          EMT      46
033414 000137 033600   JMP      74$      ;BYPASS THE REST OF THE CHECKS
033420 013737 001170 001126 70$:   MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
033426 013737 001226 001234       MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033434 113760 001226 000010       MOVVB  PORTB,RPCS2(R0) ;SELECT PORT B.
033442 005737 001164       TST     $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
033446 001414          BEQ     71$       ;BR IF ZERO
033450 013737 001224 001234       MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033456 013737 001172 001126       MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
033464 113760 001224 000010       MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
033472 005737 001166       TST     $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
033476 001012          BNE     72$       ;BR IF NOT
033500 012737 177777 001250 71$:   MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
033506 012760 000011 000000       MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
033514 012760 000013 000000       MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
033522 104026          EMT      26
033524 013737 001170 001126 72$:   MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
033532 013737 001224 001234       MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
033540 023737 001124 001170       CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
033546 001401          BEQ     73$       ;BR IF OK FROM PORT A.
033550 104007          EMT      7
033552 013737 001172 001126 73$:   MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
033560 013737 001226 001234       MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
033566 023737 001124 001172       CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
033574 001401          BEQ     74$       ;BR IF OK
033576 104007          EMT      7
033600 000240          NOP
033602 000004          74$:   NOP
                                1$:   SCOPE      ;LOOP ?
    
```

136
137

```

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

```

033604          TST26:
033604 005737 001274       TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
033610 001406          BEQ     2$         ;BR IF NOT
033612 100002          BPL     1$         ;BR IF JUST ENTERED TEST
033614 000137 003010       JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
033620 012737 177777 001274 1$:   MOV     #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
033626 112737 000026 001102 2$:   MOVVB  #26,$STNM   ;TEST NUMBER
033634 012737 033656 001106       MOV     #TEST26,$LPADR ;LOAD LOOP ON TEST ADDRESS
033642 012737 033656 001110       MOV     #TEST26,$LPERR ;LOAD LOOP ON ERROR ADDRESS
    
```

```

138 033650 012737 007640 001176      MOV    #4000.,$TIMES    ;;DO 4000. ITERATIONS
139 033656 012706 001100      TEST26: MOV    #STACK,SP    ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS

033662 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
033670 005060 000012 000012      CLR    RPDS1(R0)       ;SEIZE THE DRIVE
033674 012760 000011 000000      MOV    #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
033702 012760 000013 000000      MOV    #13,RPCS1(R0)   ;RELEASE THE DRIVE
033710 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT #B
033716 005060 000012 000012      CLR    RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
033722 012760 000011 000000      MOV    #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
033730 012760 000013 000000      MOV    #13,RPCS1(R0)   ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

033736 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
033744 013737 001224 001236      MOV    PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
033752 005060 000012 000012      CLR    RPDS1(R0)       ;WRITE RPDS1
033756 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
033764 013737 001226 001234      MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033772 013737 001226 001240      MOV    PORTB,OPPR     ;'OPPOSITE' PORT ADDRESS
034000 016037 000012 001126      MOV    RPDS1(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
034006 010037 001122 001122      MOV    R0, $BDADR      ;RH11 BASE ADDRESS
034012 062737 000012 001122      ADD    #RPDS1, $BDADR   ;GENERATE BAD REGISTER ADDRESS
034020 005037 001124 001126      CLR    $GDDAT          ;REGISTER SHOULD BE ZERO
034024 023737 001124 001126      CMP    $GDDAT, $BDDAT  ;IS THE REGISTER ZERO
034032 001403 001124 001126      BEQ    64$             ;BR IF IT IS
034034 104004 001124 001126      EMT    4
034036 000137 035010 001126      JMP    1$              ;BYPASS REST OF THE SUBTEST
034042 000137 035010 001126      64$:
034042 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
034050 013737 001224 001234      MOV    PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034056 016037 000012 001126      MOV    RPDS1(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
034064 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
034072 013737 001124 001166      MOV    $GDDAT, $TMP1    ;USE GOOD DATA AS A MASK
034100 005137 001166 001166      COM    $TMP1           ;COMPLEMENT THE EXPECTED STATUS
034104 013737 001126 001164      MOV    $BDDAT, $TMP0    ;SAVE THE ACTUAL STATUS
034112 043737 001166 001164      BIC    $TMP1, $TMP0     ;CLEAR UNWANTED BITS
034120 023737 001124 001164      CMP    $GDDAT, $TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
034126 001401 001124 001164      BEQ    65$             ;BR IF THEY ARE
034130 104005 001124 001164      EMT    5
034132 000240 001124 001164      65$: NOP

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B

034134 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
034142 013737 001226 001234      MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034150 012760 000013 000000      MOV    #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A

034156 005037 001244 000012 001126      CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
034162 016037 000012 001126      MOV    RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
034170 012737 000012 001122      MOV    #RPDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034176 060037 001122 001122      ADD    R0, $BDADR      ;ADD RH11 BASE ADDRESS
  
```

```

034202 005037 001124          CLR    $GDDAT      ;WHAT REGISTER SHOULD BE
034206 023737 001124 001126    CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
034214 001403          BEQ    66$        ;BR IF OK
034216 104010          EMT    10
034220 005137 001244          COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
034224 016037 000000 001126 66$: MOV    RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHCS1
034232 012737 000000 001122    MOV    #RPCS1,$BDADR ;FORM ADDRESS OF REGISTER
034240 060037 001122          ADD    R0,$BDADR  ;ADDRESS BASE
034244 032737 020000 001126    BIT    #MCPE,$BDDAT ;IS 'MCPE' SET ?
034252 001404          BEQ    67$        ;BR IF NOT
034254 104011          EMT    11
034256 012760 040000 000000    MOV    #TRE,RPCS1(R0) ;CLEAR 'MCPE'
034264 000240          NOP
034266 005737 001244          TST    CKERR      ;WAS RPDS1 NON ZERO ?
034272 001402          BEQ    .+6        ;CONTENTS OF RPDS1 SEEN BY PORT B
034274 000137 035010          JMP    1$        ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
  
```

;RELEASE THE DRIVE FROM PORT A

```

034300 113760 001224 000010    MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
034306 013737 001224 001234    MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034314 012760 000013 000000    MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A
  
```

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

```

034322 005037 001250          CLR    RELERR     ;CLEAR 'RELEASE ERROR' INDICATOR
034326 012737 111700 001124    MOV    #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
034334 012737 000012 001122    MOV    #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
034342 060037 001122          ADD    R0,$BDADR  ;REGISTER BASE ADDRESS FOR TYPEOUT
034346 113760 001226 000010    MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
034354 013737 001226 001234    MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034362 016037 000012 001164    MOV    RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
034370 113760 001224 000010    MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
034376 013737 001224 001234    MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034404 016037 000012 001126    MOV    RPDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT A
034412 001404          BEQ    68$        ;BR IF STATUS FROM PORT A ZERO
034414 005737 001164          TST    $TMP0     ;IS STATUS FROM PORT B ZERO ?
034420 001401          BEQ    68$        ;BR IF ZERO
034422 104031          EMT    31
034424 013737 001164 001126 68$: MOV    $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
034432 013737 001226 001234    MOV    PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
034440 023737 001124 001126    CMP    $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
034446 001401          BEQ    69$        ;BR IF OK
034450 104027          EMT    27
034452 000240          NOP
  
```

;RELEASE THE DRIVE FROM PORT B

```

034454 113760 001226 000010    MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
034462 013737 001226 001234    MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034470 012760 000013 000000    MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

034476 005037 001250          CLR    RELERR     ;CLEAR THE 'RELEASE ERROR' INDICATOR
034502 012737 000012 001122    MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
  
```

```

034510 060037 001122      ADD      R0,$BDADR      ;ADD THE I/O BASE ADDRESS
034514 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
034522 113760 001224 000010  MOV     PORTA,RPCS2(R0) ;SELECT PORT A.
034530 016037 000012 001170  MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
034536 013737 001170 001164  MOV     $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
034544 042737 100100 001164  BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034552 113760 001226 000010  MOV     PORTB,RPCS2(R0) ;SELECT PORT B.
034560 016037 000012 001172  MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
034566 013737 001172 001166  MOV     $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
034574 042737 100100 001166  BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034602 023737 001164 001166  CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
034610 001006      BNE     70$ ;BR IF NOT
034612 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
034616 001045      BNE     72$ ;BR IF NOT
034620 104046      EMT     46
034622 000137 035006      JMP     74$ ;BYPASS THE REST OF THE CHECKS
034626 013737 001170 001126 70$: MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
034634 013737 001226 001234  MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034642 113760 001226 000010  MOV     PORTB,RPCS2(R0) ;SELECT PORT B.
034650 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
034654 001414      BEQ     71$ ;BR IF ZERO
034656 013737 001224 001234  MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034664 013737 001172 001126  MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
034672 113760 001224 000010  MOV     PORTA,RPCS2(R0) ;SELECT PORT A.
034700 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
034704 001012      BNE     72$ ;BR IF NOT
034706 012737 177777 001250 71$: MCV     #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
034714 012760 000011 000000  MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
034722 012760 000013 000000  MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
034730 104026      EMT     26
034732 013737 001170 001126 72$: MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
034740 013737 001224 001234  MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
034746 023737 001124 001170  CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
034754 001401      BEQ     73$ ;BR IF OK FROM PORT A.
034756 104007      EMT     7
034760 013737 001172 001126 73$: MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
034766 013737 001226 001234  MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
034774 023737 001124 001172  CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
035002 001401      BEQ     74$ ;BR IF OK
035004 104007      EMT     7
035006 000240      74$: NOP
035010 000004      1$: SCOPE ;LOOP ?
  
```

140
155
156

```

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

157
206

```

: *
: *****
TST27:
    
```

```

035012      005737  001274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
035012      001406                BEQ      2$          ;BR IF NOT
035020      100002                BPL      1$          ;BR IF JUST ENTERED TEST
035022      000137  003010      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
035026      012737  177777  001274  1$:  MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
035034      112737  000027  001102  2$:  MOVVB   #27,$TSTNM  ;TEST NUMBER
035042      012737  035064  001106      MOV      #TEST27,$LPADR ;LOAD LOOP ON TEST ADDRESS
035050      012737  035064  001110      MOV      #TEST27,$LPERR ;LOAD LOOP ON ERROR ADDRESS
035056      012737  007640  001176      MOV      #4000,$TIMES  ;DO 4000. ITERATIONS
157 035064  012706  001100  TEST27: MOV     #STACK,SP ;LOAD THE STACK POINTER
    
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

035070      113760  001224  000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT #A
035076      005060  000012                CLR      RPDS1(R0)      ;SEIZE THE DRIVE
035102      012760  000011  000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
035110      012760  000013  000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
035116      113760  001226  000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT #B
035124      005060  000012                CLR      RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
035130      012760  000011  000000      MOV      #11,RPCS1(R0)  ;ISSUE DRIVE CLEAR
035136      012760  000013  000000      MOV      #13,RPCS1(R0)  ;RELEASE THE DRIVE
    
```

;SELECT DRIVE OTHER THAN THAT BEING TESTED

```

035144      113760  001230  000010      MOVVB   PORTC,RPCS2(R0) ;SELECT DRIVE NOT BEING TESTED
035152      013737  001224  001236      MOV      PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
    
```

;WRITE THE DRIVE'S ATTENTION BIT

```

035160      013760  001232  000016      MOV      ASR1,RPAS(R0)  ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
035166      113760  001224  000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
035174      013737  001224  001234      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

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

```

035202      005760  000012                TST      RPDS1(R0)      ;SEE THE REGISTER THROUGH PORT A ?
035206      001014                BNE     1$              ;BR IF YES
035210      113760  001226  000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B
035216      013737  001226  001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035224      005760  000012                TST      RPDS1(R0)      ;SEE REGISTER THROUGH PORT B ?
035230      001021                BNE     2$              ;BR IF YES
035232      104042                EMT     42
035234      000137  037004                JMP     4$              ;BYPASS REST OF TEST
035240      113760  001226  000010  1$:  MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B
035246      013737  001226  001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035254      005760  000012                TST      RPDS1(R0)      ;REGISTER SHOULD BE ZERO THROUGH PORT B
035260      001002                BNE     +6              ;BR IF STATUS REG IS NOT ZERO
035262      000137  036134                JMP     3$              ;STATUS REG IS ZERO
035266      104043                EMT     43
035270      000137  037004                JMP     4$              ;BYPASS REST OF TEST
    
```

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

```

035274          2$:
035274 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
035300 016037 000012 001126  MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
035306 012737 000012 001122  MOV      #RPDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
035314 060037 001122          ADD      RO, $BDADR      ;ADD RH11 BASE ADDRESS
035320 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
035326 013737 001126 001164  MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
035334 042737 106077 001164  BIC      #^C71700, $TMP0 ;SAVE SPECIFIED BITS
035342 023737 001124 001164  CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
035350 001414          BEQ      64$           ;BR IF OK
035352 013737 001126 001174  MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
035360 042737 071700 001174  BIC      #71700, $TMP4   ;CLEAR THE MASKED BITS
035366 053737 001174 001124  BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
035374 104010          EMT      10
035376 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
035402 000240          64$: NOP
035404 013737 001226 001236  MOV      PORTB, SEIZPT   ;ADDRESS FOR ERROR MESSAGE
035412 013737 001224 001240  MOV      PORTA, OPPRT    ;SAME AS ABOVE

;RELEASE THE DRIVE FROM PORT B

035420 113760 001226 000010  MOVB     PORTB, RPCS2(RO) ;SELECT PORT B
035426 013737 001226 001234  MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035434 012760 000013 000000  MOV      #13, RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT B

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

035442 005037 001250          CLR      RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
035446 012737 111700 001124  MOV      #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
035454 012737 000012 001122  MOV      #RPDS1, $BDADR  ;REGISTER ADDRESS INCREMENT
035462 060037 001122          ADD      RO, $BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
035466 113760 001224 000010  MOVB     PORTA, RPCS2(RO) ;SELECT PORT A
035474 013737 001224 001234  MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035502 016037 000012 001164  MOV      RPDS1(RO), $TMP0 ;READ STATUS REGISTER FROM PORT A
035510 113760 001226 000010  MOVB     PORTB, RPCS2(RO) ;SELECT PORT B
035516 013737 001226 001234  MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035524 016037 000012 001126  MOV      RPDS1(RO), $BDDAT ;DRIVE STATUS FROM PORT B
035532 001404          BEQ      66$           ;BR IF STATUS FROM PORT B ZERO
035534 005737 001164          TST      $TMP0          ;IS STATUS FROM PORT A ZERO ?
035540 001401          BEQ      66$           ;BR IF ZERO
035542 104044          EMT      44
035544 013737 001164 001126  66$: MOV      $TMP0, $BDDAT   ;CHECK STATUS FROM PORT A
035552 013737 001224 001234  MOV      PORTA, PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
035560 023737 001124 001126  CMP      $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
035566 001401          BEQ      67$           ;BR IF OK
035570 104027          EMT      27
035572 000240          67$: NOP

;RELEASE THE DRIVE FROM PORT A

035574 113760 001224 000010  MOVB     PORTA, RPCS2(RO) ;SELECT PORT A
035602 013737 001224 001234  MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035610 012760 000013 000000  MOV      #13, RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

035616 005037 001250          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
  
```

```

035622 012737 000012 001122      MOV      #RPDS1,$BDADR      ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
035630 060037 001122              ADD      R0,$BDADR        ;ADD THE I/O BASE ADDRESS
035634 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
035642 113760 001224 000010      MOVVB   PORTA,RPCS2(R0)   ;SELECT PORT A.
035650 016037 000012 001170      MOV      RPDS1(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
035656 013737 001170 001164      MOV      $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
035664 042737 100100 001164      BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035672 113760 001226 000010      MOVVB   PORTB,RPCS2(R0)   ;SELECT PORT B.
035700 016037 000012 001172      MOV      RPDS1(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
035706 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
035714 042737 100100 001166      BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035722 023737 001164 001166      CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
035730 001006                      BNE     68$              ;BR IF NOT
035732 005737 001164              TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
035736 001045                      BNE     70$              ;BR IF NOT
035740 104046                      EMT     46
035742 000137 036126              JMP     72$              ;BYPASS THE REST OF THE CHECKS
035746 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
035754 013737 001226 001234      MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035762 113760 001226 000010      MOVVB   PORTB,RPCS2(R0)   ;SELECT PORT B.
035770 005737 001164              TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
035774 001414                      BEQ     69$              ;BR IF ZERO
035776 013737 001224 001234      MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
036004 013737 001172 001126      MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
036012 113760 001224 000010      MOVVB   PORTA,RPCS2(R0)   ;SELECT PORT A.
036020 005737 001166              TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
036024 001012                      BNE     70$              ;BR IF NOT
036026 012737 177777 001250 69$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
036034 012760 000011 000000      MOV     #11,RPCS1(R0)    ;CLEAR THE DRIVE
036042 012760 000013 000000      MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
036050 104026                      EMT     26
036052 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
036060 013737 001224 001234      MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
036066 023737 001124 001170      CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
036074 001401                      BEQ     71$              ;BR IF OK FROM PORT A.
036076 104007                      EMT     7
036100 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
036106 013737 001226 001234      MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
036114 023737 001124 001172      CMP     $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
036122 001401                      BEQ     72$              ;BR IF OK
036124 104007                      EMT     7
036126 000240 72$:  NOP
036130 000137 037004              JMP     4$
    
```

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

```

036134 113760 001224 000010 3$:  MOVVB   PORTA,RPCS2(R0)   ;SELECT PORT A
036142 013737 001224 001234      MOV     PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036150 005037 001244              CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
036154 016037 000012 001126      MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
036162 012737 000012 001122      MOV     #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036170 060037 001122              ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
036174 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
036202 013737 001126 001164      MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
036210 042737 106077 001164      BIC     #^C71700,$TMP0   ;SAVE SPECIFIED BITS
036216 023737 001124 001164      CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
    
```



```

036224 001414          BEQ      73$          ;BR IF OK
036226 013737 001126 001174  MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
036234 042737 071700 001174  BIC      #71700,$STMP4 ;CLEAR THE MASKED BITS
036242 053737 001174 001124  BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
036250 104010          EMT      10
036252 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
036256 000240          NOP
036260 013737 001224 001236 73$:  MOV      PORTA,SEIZPT ;ADDRESS FOR ERROR MESSAGE
036266 013737 001226 001240  MOV      PORTB,OPPRT ;SAME AS ABOVE

;RELEASE THE DRIVE FROM PORT A

036274 113760 001224 000010  MOVB     PORTA,RPCS2(R0) ;SELECT PORT A
036302 013737 001224 001234  MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036310 012760 000013 000000  MOV      #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

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

036316 005037 001250          CLR      RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
036322 012737 111700 001124  MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
036330 012737 000012 001122  MOV      #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
036336 060037 001122          ADD      R0,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
036342 113760 001226 000010  MOVB     PORTB,RPCS2(R0) ;SELECT PORT B
036350 013737 001226 001234  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036356 016037 000012 001164  MOV      RPDS1(R0),$STMP0 ;READ STATUS REGISTER FROM PORT B
036364 113760 001224 000010  MOVB     PORTA,RPCS2(R0) ;SELECT PORT A
036372 013737 001224 001234  MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036400 016037 000012 001126  MOV      RPDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT A
036406 001404          BEQ      75$          ;BR IF STATUS FROM PORT A ZERO
036410 005737 001164          TST      $STMP0          ;IS STATUS FROM PORT B ZERO ?
036414 001401          BEQ      75$          ;BR IF ZERO
036416 104044          EMT      44
036420 013737 001164 001126 75$:  MOV      $STMP0,$BDDAT ;CHECK STATUS FROM PORT B
036426 013737 001226 001234  MOV      PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
036434 023737 001124 001126  CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
036442 001401          BEQ      76$          ;BR IF OK
036444 104027          EMT      27
036446 000240          NOP

76$:

;RELEASE THE DRIVE FROM PORT B

036450 113760 001226 000010  MOVB     PORTB,RPCS2(R0) ;SELECT PORT B
036456 013737 001226 001234  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036464 012760 000013 000000  MOV      #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

036472 005037 001250          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
036476 012737 000012 001122  MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
036504 060037 001122          ADD      R0,$BDADR ;ADD THE I/O BASE ADDRESS
036510 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
036516 113760 001224 000010  MOVB     PORTA,RPCS2(R0) ;SELECT PORT A.
036524 016037 000012 001170  MOV      RPDS1(R0),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
036532 013737 001170 001164  MOV      $STMP2,$STMP0 ;COPY IT INTO 'STMP0'
036540 042737 100100 001164  BIC      #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
036546 113760 001226 000010  MOVB     PORTB,RPCS2(R0) ;SELECT PORT B.
036554 016037 000012 001172  MOV      RPDS1(R0),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.

```

```

036562 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
036570 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
036576 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
036604 001006                                     BNE      77$              ;BR IF NOT
036606 005737 001164                                     TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
036612 001045                                     BNE      79$              ;BR IF NOT
036614 104046                                     EMT      46
036616 000137 037002                                     JMP      81$              ;BYPASS THE REST OF THE CHECKS
036622 013737 001170 001126 77$:   MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
036630 013737 001226 001234       MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
036636 113760 001226 000010       MOVVB   PORTB,RPCS2(R0)  ;SELECT PORT B.
036644 005737 001164                                     TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
036650 001414                                     BEQ      78$              ;BR IF ZERO
036652 013737 001224 001234       MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
036660 013737 001172 001126       MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
036666 113760 001224 000010       MOVVB   PORTA,RPCS2(R0)  ;SELECT PORT A.
036674 005737 001166                                     TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
036700 001012                                     BNE      79$              ;BR IF NOT
036702 012737 177777 001250 78$:   MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
036710 012760 000011 000000       MOV      #11,RPCS1(R0)   ;CLEAR THE DRIVE
036716 012760 000013 000000       MOV      #13,RPCS1(R0)   ;RELEASE THE DRIVE
036724 104026                                     EMT      26
036726 013737 001170 001126 79$:   MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
036734 013737 001224 001234       MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
036742 023737 001124 001170       CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
036750 001401                                     BEQ      80$              ;BR IF OK FROM PORT A.
036752 104007                                     EMT      7
036754 013737 001172 001126 80$:   MOV      $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
036762 013737 001226 001234       MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
036770 023737 001124 001172       CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
036776 001401                                     BEQ      81$              ;BR IF OK
037000 104007                                     EMT      7
037002 000240 81$:   NOP
037004 000004 4$:   SCOPE                    ;LOOP ?

```

207
219
220

```

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

```

```

037006 005737 001274      TST30:  TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
037006 001406                                     BEQ      2$              ;BR IF NOT
037012 100002                                     BPL      1$              ;BR IF JUST ENTERED TEST
037016 000137 003010       JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
037022 012737 177777 001274 1$:   MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
037030 112737 000030 001102 2$:   MOVVB   #30,$STNM       ;TEST NUMBER
037036 012737 037060 001106       MOV      #TEST30,$LPADR  ;LOAD LOOP ON TEST ADDRESS
037044 012737 037060 001110       MOV      #TEST30,$LPERR  ;LOAD LOOP ON ERROR ADDRESS

```

221
237

```

037052 012737 007640 001176      MOV    #4000, $TIMES      ;; DO 4000. ITERATIONS
037060 012706 001100      TEST30: MOV    #STACK, SP  ;; LOAD THE STACK POINTER

                                ; CLEAR ATTENTION BITS FOR BOTH PORTS

037064 113760 001224 000010      MOVB   PORTA, RPCS2(R0)  ;; SELECT PORT #A
037072 005060 000012                CLR    RPDS1(R0)         ;; SEIZE THE DRIVE
037076 012760 000011 000000      MOV    #11, RPCS1(R0)   ;; ISSUE DRIVE CLEAR
037104 012760 000013 000000      MCV   #13, RPCS1(R0)   ;; RELEASE THE DRIVE
037112 113760 001226 000010      MCVB  PORTB, RPCS2(R0)  ;; SELECT PORT #B
037120 005060 000012                CLR    RPDS1(R0)         ;; SEIZE THE DRIVE THROUGH PORT 'B'
037124 012760 000011 000000      MOV    #11, RPCS1(R0)   ;; ISSUE DRIVE CLEAR
037132 012760 000013 000000      MOV    #13, RPCS1(R0)   ;; RELEASE THE DRIVE
037140 113760 001230 000010      MOVB  PORTC, RPCS2(R0)  ;; SELECT DRIVE NOT BEING TESTED

                                ; WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST

037146 013737 001232 001164      MOV    ASR1, $TMP0      ;; STORE ATTN BIT FOR PORT A
037154 005137 001164                COM   $TMP0             ;; COMPLEMENT IT
037160 013760 001164 000016      MOV    $TMP0, RPAS(R0)  ;; WRITE THE ATTN REGISTER

                                ; VERIFY THAT DRIVE REMAINED IN NEUTRAL

                                ; VERIFY THAT THE DRIVE IS IN NEUTRAL

037166 005037 001250                CLR    RELERR           ;; CLEAR THE 'RELEASE ERROR' INDICATOR
037172 012737 000012 001122      MOV    #RPDS1, $BDADR   ;; FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
037200 060037 001122                ADD   RO, $BDADR       ;; ADD THE I/O BASE ADDRESS
037204 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;; COMPARISON CONSTANT
037212 113760 001224 000010      MOVB  PORTA, RPCS2(R0)  ;; SELECT PORT A.
037220 016037 000012 001170      MOV    RPDS1(R0), $TMP2 ;; GET THE DRIVE STATUS REGISTER FROM PORT A.
037226 013737 001170 001164      MOV    $TMP2, $TMP0     ;; COPY IT INTO '$TMP0'
037234 042737 100100 001164      BIC   #ATA!VV, $TMP0    ;; CLEAR PORT DEPENDENT BITS FROM THE COPY
037242 113760 001226 000010      MOVB  PORTB, RPCS2(R0)  ;; SELECT PORT B.
037250 016037 000012 001172      MOV    RPDS1(R0), $TMP3 ;; GET THE DRIVE STATUS REGISTER FROM PORT B.
037256 013737 001172 001166      MOV    $TMP3, $TMP1    ;; COPY IT INTO '$TMP1'
037264 042737 100100 001166      BIC   #ATA!VV, $TMP1    ;; CLEAR PORT DEPENDENT BITS FROM THE COPY
037272 023737 001164 001166      CMP   $TMP0, $TMP1     ;; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
037300 001006                BNE   64$              ;; BR IF NOT
037302 005737 001164                TST  $TMP0             ;; REGISTERS ARE THE SAME: ARE THEY ZERO ?
037306 001045                BNE   66$              ;; BR IF NOT
037310 104046                EMT   46
037312 000137 037476                JMP   68$              ;; BYPASS THE REST OF THE CHECKS
037316 013737 001170 001126 64$: MOV    $TMP2, $BDDAT    ;; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
037324 013737 001226 001234      MOV    PORTB, PTNBR     ;; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037332 113760 001226 000010      MOVB  PORTB, RPCS2(R0)  ;; SELECT PORT B.
037340 005737 001164                TST  $TMP0             ;; SEE IF STATUS EQ 0 FROM PORT A.
037344 001414                BEQ   65$              ;; BR IF ZERO
037346 013737 001224 001234      MOV    PORTA, PTNBR     ;; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037354 013737 001172 001126      MOV    $TMP3, $BDDAT    ;; 'BAD DATA' FOR ERROR TYPE OUT
037362 113760 001224 000010      MOVB  PORTA, RPCS2(R0)  ;; SELECT PORT A.
037370 005737 001166                TST  $TMP1             ;; SEE IF STATUS EQ ZERO FROM PORT B.
037374 001012                BNE   66$              ;; BR IF NOT
037376 012737 177777 001250 65$: MOV    #-1, RELERR     ;; SET 'RELEASE ERROR' INDICATOR
037404 012760 000011 000000      MOV    #11, RPCS1(R0)  ;; CLEAR THE DRIVE
037412 012760 000013 000000      MOV    #13, RPCS1(R0)  ;; RELEASE THE DRIVE
    
```

```

037420 104021          EMT      21
037422 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
037430 013737 001224 001234      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
037436 023737 001124 001170      CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
037444 001401          BEQ     67$          ;BR IF OK FROM PORT A.
037446 104007          EMT      7
037450 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
037456 013737 001226 001234      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
037464 023737 001124 001172      CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
037472 001401          BEQ     68$          ;BR IF OK
037474 104007          EMT      7
037476 000240          68$:  NOP
037500 000004          SCOPE          ;LOOP ?
    
```

238
252
253

```

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

254
302

```

037502 005737 001274          TST31: TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
037502 001406          BEQ     2$          ;BR IF NOT
037510 100002          BPL     1$          ;BR IF JUST ENTERED TEST
037512 000137 003010          JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
037516 012737 177777 001274 1$:  MOV     #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
037524 112737 000031 001102 2$:  MOVB   #31,$STNM   ;TEST NUMBER
037532 012737 037554 001106      MOV     #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
037540 012737 037554 001110      MOV     #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
037546 012737 000004 001176      MOV     #4,$TIMES   ;DO 4 ITERATIONS
037554 012706 001100          TEST31: MOV     #STACK,SP  ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS

037560 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
037566 005060 000012          CLR     RPDS1(R0)     ;SEIZE THE DRIVE
037572 012760 000011 000000      MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
037600 012760 000013 000000      MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
037606 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT #B
037614 005060 000012          CLR     RPDS1(R0)     ;SEIZE THE DRIVE THROUGH PORT 'B'
037620 012760 000011 000000      MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
037626 012760 000013 000000      MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

037634 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
037642 013737 001224 001236      MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
037650 005060 000012          CLR     RPDS1(R0)     ;WRITE RPDS1
    
```

```

037654 013737 001226 001240      MOV     PORTB,OPPRT      ;'OPPOSITE' PORT ADDRESS
                                ;FORCE AN ERROR

037662 012760 177777 000014      MOV     #-1,RPER1(R0)  ;SET ERROR BITS
                                ;START THE TIMER

037670 005037 001252              CLR     TIME             ;CLEAR THE ELAPSED TIME COUNTER
037674 012737 003720 001254      MOV     #2000.,WATCH   ;SET WATCH TO 2000 MS
037702 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
037710 013737 001226 001234      MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT

                                ;WAIT FOR DRIVE TO TIMEOUT

037716 005760 000012      1$:    TST     RPDS1(R0)   ;WAIT FOR THE DRIVE TO BE RELEASED
037722 001004              BNE     2$              ;BR IF DRIVE RELEASED
037724 005737 001254      TST     WATCH          ;WATCH AT ZERO ?
037730 001372              BNE     1$              ;BR IF NOT
037732 104036              EMT     36
037734              2$:
037734 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
037742 013737 001224 001234      MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT

                                ;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET

037750 005037 001244              CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
037754 016037 000012 001126      MOV     RPDS1(R0),SBDAT ;GET CONTENTS OF RPDS1
037762 012737 000012 001122      MOV     #RPDS1,SBDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037770 060037 001122              ADD     R0,SBDADR      ;ADD RH11 BASE ADDRESS
037774 012737 040000 001124      MOV     #ERR,$GDDAT    ;WHAT REGISTER SHOULD BE
040002 013737 001126 001164      MOV     SBDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
040010 042737 137777 001164      BIC     #^C40000,$TMP0 ;SAVE SPECIFIED BITS
040016 023737 001124 001164      CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
040024 001414              BEQ     66$            ;BR IF OK
040026 013737 001126 001174      MOV     SBDAT,$TMP4    ;COPY 'BAD DATA'
040034 042737 040000 001174      BIC     #40000,$TMP4   ;CLEAR THE MASKED BITS
040042 053737 001174 001124      BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TIMEOUT
040050 104023              EMT     23
040052 005137 001244              COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
040056 000240      66$:    NOP

                                ;THE ERROR REGISTER SHOULD CONTAIN 1'S

040060 005037 001244              CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040064 016037 000014 001126      MOV     RPER1(R0),SBDAT ;GET CONTENTS OF RPER1
040072 012737 000014 001122      MOV     #RPER1,SBDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040100 060037 001122              ADD     R0,SBDADR      ;ADD RH11 BASE ADDRESS
040104 012737 177777 001124      MOV     #177777,$GDDAT ;WHAT REGISTER SHOULD BE
040112 023737 001124 001126      CMP     $GDDAT,$SBDAT  ;IS THE REGISTER OK ?
040120 001403              BEQ     68$            ;BR IF OK
040122 104010              EMT     10
040124 005137 001244              COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
040130 000240      68$:    NOP

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

```

040132 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
040136 016037 000012 001126  MOV      RPDS1(R0),%BDDAT ;GET CONTENTS OF RPDS1
040144 012737 000012 001122  MOV      %RPDS1,%BADDR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040152 060037 001122          ADD      R0,%BADDR      ;ADD RH11 BASE ADDRESS
040156 012737 100000 001124  MOV      %ATA,%GDDAT    ;WHAT REGISTER SHOULD BE
040164 013737 001126 001164  MOV      %BDDAT,%STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
040172 042737 077777 001164  BIC      %ATA,%STMP0    ;SAVE SPECIFIED BITS
040200 023737 001124 001164  CMP      %GDDAT,%STMP0  ;COMPARE THE BITS
040206 001414          BEQ      70$           ;BR IF OK
040210 013737 001126 001174  MOV      %BDDAT,%STMP4  ;COPY 'BAD DATA'
040216 042737 100000 001174  BIC      %ATA,%STMP4    ;CLEAR THE MASKED BITS
040224 053737 001174 001124  BIS      %STMP4,%GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
040232 104041          EMT      41
040234 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
040240 000240          70$: NOP
  
```

:VERIFY THAT THE DRIVE IS IN NEUTRAL

```

040242 005037 001250          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
040246 012737 000012 001122  MOV      %RPDS1,%BADDR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
040254 060037 001122          ADD      R0,%BADDR     ;ADD THE I/O BASE ADDRESS
040260 012737 051700 001124  MOV      %51700,%GDDAT ;COMPARISON CONSTANT
040266 113760 001224 000010  MOVB    PORTA,%RPCS2(R0) ;SELECT PORT A.
040274 016037 000012 001170  MOV      RPDS1(R0),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
040302 013737 001170 001164  MOV      %STMP2,%STMP0  ;COPY IT INTO 'STMP0'
040310 042737 100100 001164  BIC      %ATA!VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040316 113760 001226 000010  MOVB    PORTB,%RPCS2(R0) ;SELECT PORT B.
040324 016037 000012 001172  MOV      RPDS1(R0),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
040332 013737 001172 001166  MOV      %STMP3,%STMP1 ;COPY IT INTO 'STMP1'
040340 042737 100100 001166  BIC      %ATA!VV,%STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040346 023737 001164 001166  CMP      %STMP0,%STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
040354 001006          BNE      72$           ;BR IF NOT
040356 005737 001164          TST      %STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
040362 001045          BNE      74$           ;BR IF NOT
040364 104046          EMT      46
040366 000137 040566          JMP      76$           ;BYPASS THE REST OF THE CHECKS
040372 013737 001170 001126 72$: MOV      %STMP2,%BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
040400 013737 001226 001234  MOV      PORTB,%PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040406 113760 001226 000010  MOVB    PORTB,%RPCS2(R0) ;SELECT PORT B.
040414 005737 001164          TST      %STMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
040420 001414          BEQ      73$           ;BR IF ZERO
040422 013737 001224 001234  MOV      PORTA,%PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040430 013737 001172 001126  MOV      %STMP3,%BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
040436 113760 001224 000010  MOVB    PORTA,%RPCS2(R0) ;SELECT PORT A.
040444 005737 001166          TST      %STMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
040450 001012          BNE      74$           ;BR IF NOT
040452 012737 177777 001250 73$: MOV      #-1,%RELERR    ;SET 'RELEASE ERROR' INDICATOR
040460 012760 000011 000000  MOV      #11,%RPCS1(R0) ;CLEAR THE DRIVE
040466 012760 000013 000000  MOV      #13,%RPCS1(R0) ;RELEASE THE DRIVE
040474 104026          EMT      26
040476 013737 001170 001126 74$: MOV      %STMP2,%BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
040504 013737 001224 001234  MOV      PORTA,%PTNBR   ;CHANGE PORT NUMBER
040512 042737 100000 001170  BIC      %ATA,%STMP2    ;DON'T CHECK THE ATTN BIT
040520 023737 001124 001170  CMP      %GDDAT,%STMP2 ;ALL BITS OK ?
040526 001401          BEQ      75$           ;BR IF OK FROM PORT A.
  
```

```

040530 104007          EMT      7
040532 013737 001172 001126 75$:  MOV    $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
040540 013737 001226 001234      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
040546 042737 100000 001172      BIC    #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
040554 023737 001124 001172      CMP    $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
040562 001401          BEQ    76$ ;BR IF OK
040564 104007          EMT      7
040566 000240          76$:  NOP
    
```

;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

```

040570 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
040576 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040604 005037 001244              CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
040610 016037 000012 001126      MOV    RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
040616 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040624 060037 001122              ADD    R0,$BDADR ;ADD RH11 BASE ADDRESS
040630 005037 001124              CLR    $GDDAT ;WHAT REGISTER SHOULD BE
040634 013737 001126 001164      MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
040642 042737 077777 001164      BIC    #^CATA,$TMP0 ;SAVE SPECIFIED BITS
040650 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
040656 001414          BEQ    77$ ;BR IF OK
040660 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
040666 042737 100000 001174      BIC    #ATA,$TMP4 ;CLEAR THE MASKED BITS
040674 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
040702 104052          EMT      52
040704 005137 001244          COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
040710 000240          77$:  NOP
    
```

;CLEAR ATTENTION BIT FOR PORT A

```

040712 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
040720 005060 000012              CLR    RPDS1(R0) ;SEIZE THE DRIVE
040724 012760 000011 000000      MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
040732 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
040740 000004          3$:  SCOPE ;LOOP ?
    
```

316
317

```

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

```

040742          TST32:
040742 005737 001274          TST    KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
040746 001406          JEQ    2$ ;BR IF NOT
040750 100002          BPL    1$ ;BR IF JUST ENTERED TEST
040752 000137 003010          JMP    EXEC ;RETURN & GET NEXT TEST NUMBER
040756 012737 177777 001274 1$:  MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
    
```

```

040764 112737 000032 0J1102 2$:   MOVB   #32,$STSTM   ;TEST NUMBER
040772 012737 041014 001106   MOV   #TEST32,$LPADR ;LOAD LOOP ON TEST ADDRESS
041000 012737 041014 001110   MOV   #TEST32,$LPERR ;LOAD LOOP ON ERROR ADDRESS
041006 012737 000004 001176   MOV   #4,$TIMES     ;DO 4 ITERATIONS
318 041014 012706 001100 TEST32: MOV   #STACK,SP   ;LOAD THE STACK POINTER
319
;CLEAR ATTENTION BITS FOR BOTH PORTS

041020 113760 001224 000010   MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
041026 005060 000012   CLR   RPDS1(R0)       ;SEIZE THE DRIVE
041032 012760 000011 000000   MOV   #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
041040 012760 000013 000000   MOV   #13,RPCS1(R0)   ;RELEASE THE DRIVE
041046 113760 001226 000010   MOVB   PORTB,RPCS2(R0) ;SELECT PORT #B
041054 005060 000012   CLR   RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
041060 012760 000011 000000   MOV   #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
041066 012760 000013 000000   MOV   #13,RPCS1(R0)   ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

041074 113760 001226 000010   MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
041102 013737 001226 001236   MOV   PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
041110 005060 000012   CLR   RPDS1(R0)       ;WRITE RPDS1
041114 013737 001224 001240   MOV   PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

;FORCE AN ERROR

041122 012760 177777 000014   MOV   #-1,RPER1(R0)   ;SET ERROR BITS

;START THE TIMER

041130 005037 001252   CLR   TIME            ;CLEAR THE ELAPSED TIME COUNTER
041134 012737 003720 001254   MOV   #2000, WATCH    ;SET WATCH TO 2000 MS
041142 113760 001224 000010   MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
041150 013737 001224 001234   MOV   PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;WAIT FOR DRIVE TO TIMEOUT

041156 005760 000012 1$:   TST   RPDS1(R0)       ;WAIT FOR THE DRIVE TO BE RELEASED
041162 001004   BNE   2$              ;BR IF DRIVE RELEASED
041164 005737 001254   TST   WATCH           ;WATCH AT ZERO ?
041170 001372   BNE   1$              ;BR IF NOT
041172 104036   EMT   36
041174
041174 113760 001226 000010 2$:   MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
041202 013737 001226 001234   MOV   PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET

041210 005037 001244   CLR   CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
041214 016037 000012 001126   MOV   RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
041222 012737 000012 001122   MOV   #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041230 060037 001122   ADD   R0,$BDADR       ;ADD RH11 BASE ADDRESS
041234 012737 040000 001124   MOV   #ERR,$GDDAT     ;WHAT REGISTER SHOULD BE
041242 013737 001126 001164   MOV   $BDDAT,$TMPO    ;MOVE REGISTER CONTENTS TO 'TMPO'
041250 042737 137777 001164   BIC   #^C40000,$TMPO  ;SAVE SPECIFIED BITS
041256 023737 001124 001164   CMP   $GDDAT,$TMPO    ;COMPARE THE BITS
041264 001414   BEQ   66$            ;BR IF OK
    
```



```

041266 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
041274 042737 040000 001174      BIC      #40000,$TMP4     ;CLEAR THE MASKED BITS
041302 053737 001174 001126      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
041310 104023                      EMT      23
041312 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
041316 000240                      66$:    NOP
    
```

;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

041320 005037 001244                      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
041324 016037 000014 001126      MOV      RPER1(RO),$BDDAT ;GET CONTENTS OF RPER1
041332 012737 000014 001122      MOV      #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041340 060037 001122                      ADD      RO,$BDADR        ;ADD RH11 BASE ADDRESS
041344 012737 177777 001124      MOV      #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
041352 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
041360 001403                      BEQ      68$              ;BR IF OK
041362 104010                      EMT      10
041364 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
041370 000240                      68$:    NOP
    
```

;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

```

041372 005037 001244                      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
041376 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
041404 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041412 060037 001122                      ADD      RO,$BDADR        ;ADD RH11 BASE ADDRESS
041416 012737 100000 001124      MOV      #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
041424 013737 001126 001164      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
041432 042737 077777 001164      BIC      #^CATA,$TMP0    ;SAVE SPECIFIED BITS
041440 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
041446 001414                      BEQ      70$              ;BR IF OK
041450 013737 001126 001174      MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
041456 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
041464 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
041472 104041                      EMT      41
041474 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
041500 000240                      70$:    NOP
    
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

041502 005037 001250                      CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
041506 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
041514 060037 001122                      ADD      RO,$BDADR        ;ADD THE I/O BASE ADDRESS
041520 012737 051700 001124      MOV      #51700,$GDDAT   ;COMPARISON CONSTANT
041526 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
041534 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
041542 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
041550 042737 100100 001164      BIC      #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
041556 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
041564 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
041572 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
041600 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
041606 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
041614 001006                      BNE      72$              ;BR IF NOT
041616 005737 001164                      TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
    
```

```

041622 001045          BNE      74$          ;BR IF NOT
041624 104046          EMT      46
041626 000137 042026   JMP      76$          ;BYPASS THE REST OF THE CHECKS
041632 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
041640 013737 001226 001234   MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
041646 113760 001226 000010   MOVVB  PORTB,RPCS2(R0) ;SELECT PORT B.
041654 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
041660 001414          BEQ     73$          ;BR IF ZERO
041662 013737 001224 001234   MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
041670 013737 001172 001126   MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
041676 113760 001224 000010   MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
041704 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
041710 001012          BNE     74$          ;BR IF NOT
041712 012737 177777 001250 73$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
041720 012760 000011 000000   MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
041726 012760 000013 000000   MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
041734 104026          EMT      26
041736 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
041744 013737 001224 001234   MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
041752 042737 100000 001170   BIC     #ATA,$TMP2   ;DON'T CHECK THE ATTN BIT
041760 023737 001124 001170   CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
041766 001401          BEQ     75$          ;BR IF OK FROM PORT A.
041770 104007          EMT      7
041772 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
042000 013737 001226 001234   MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
042006 042737 100000 001172   BIC     #ATA,$TMP3   ;DON'T CHECK THE ATTN BIT
042014 023737 001124 001172   CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
042022 001401          BEQ     76$          ;BR IF OK
042024 104007          EMT      7
042026 000240          NOP

```

;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

```

042030 113760 001224 000010   MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A
042036 013737 001224 001234   MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042044 005037 001244          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
042050 016037 000012 001126   MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
042056 012737 000012 001122   MOV     #RPDS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042064 060037 001122          ADD     R0,$BADDR    ;ADD RH11 BASE ADDRESS
042070 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
042074 013737 001126 001164   MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
042102 042737 077777 001164   BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
042110 023737 001124 001164   CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
042116 001414          BEQ     77$          ;BR IF OK
042120 013737 001126 001174   MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
042126 042737 100000 001174   BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
042134 053737 001174 001124   BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
042142 104052          EMT     52
042144 005137 001244          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
042150 000240          NOP

```

;CLEAR ATTENTION BIT FOR PORT B

```

042152 113760 001226 000010   MOVVB  PORTB,RPCS2(R0) ;SELECT PORT #B
042160 005060 000012          CLR     RPDS1(R0)    ;SEIZE THE DRIVE
042164 012760 000011 000000   MOV     #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
042172 012760 000013 000000   MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE

```

042200 000004 3\$: SCOPE ;LOOP ?
 320
 340
 341

```

*****
*TEST 33      TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
*
*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
*  BITS ARE SET IN THE DRIVE.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
*  B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
*
*  C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO'
*     BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
*     THAT RPER1 HAS NOT BEEN CLEARED.
*
*  D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
*
*  E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
*     RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****
    
```

```

042202
042202 005737 001274
042206 001406
042210 100002
042212 000137 003010
042216 012737 177777 001274 1$:
042224 112737 000033 001102 2$:
042232 012737 042254 001106
042240 012737 042254 001110
042246 012737 007640 001176
342 042254 012706 001100
374

TST33:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOVVB   #33,$TSTNM ;TEST NUMBER
          MOV      #TEST33,$LPADR ;LOAD LOOP ON TEST ADDRESS
          MOV      #TEST33,$LPERR ;LOAD LOOP ON ERROR ADDRESS
          MOV      #4000,$TIMES ;DO 4000. ITERATIONS
TEST33:  MOV      #STACK,SP ;LOAD THE STACK POINTER

          ;CLEAR ATTENTION BITS FOR BOTH PORTS
          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

          ;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

          ;FORCE AN ERROR
          MOV     #-1,RPER1(R0)   ;SET ERROR BITS
          MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
    
```

```

042376 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042402 016037 000000 001126 MOV RPCS1(R0), $BDDAT ;GET CONTENTS OF RPCS1
042410 012737 000000 001122 MOV #RPCS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042416 060037 001122 ADD R0, $BDADR ;ADD RH11 BASE ADDRESS
042422 012737 004012 001124 MOV #4012, $GDDAT ;WHAT REGISTER SHOULD BE
042430 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
042436 042737 173765 001164 BIC #4012, $TMP0 ;SAVE SPECIFIED BITS
042444 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
042452 001414 BEQ 66$ ;BR IF OK
042454 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
042462 042737 004012 001174 BIC #4012, $TMP4 ;CLEAR THE MASKED BITS
042470 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
042476 104025 EMT 25
042500 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
042504 000240 66$: NOP
042506 005737 001244 TST CKERR ;DID 'GO' BIT RESET ?
042512 001002 BNE .+6 ;BR IF NOT
042514 000137 042554 JMP 1$ ;'GO' BIT RESET
042520 012760 000040 000010 MOV #CLR, RPCS2(R0) ;INIT THE RH11
042526 113760 001224 000010 MOVB PORTA, RPCS2(R0) ;SELECT PORT A
042534 013737 001224 001234 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042542 012760 000013 000000 MOV #13, RPCS1(R0) ;RELEASE THE DRIVE THROUGH PORT A
042550 000137 043270 JMP 2$ ;BYPASS THE REST OF THE TEST
  
```

;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A

```

042554 113760 001226 000010 1$: MOVB PORTB, RPCS2(R0) ;SELECT PORT B
042554 113760 001226 000010 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042562 013737 001226 001234 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042570 005037 001244 MOV RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
042574 016037 000012 001126 MOV #RPDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042602 012737 000012 001122 ADD R0, $BDADR ;ADD RH11 BASE ADDRESS
042610 060037 001122 CLR $GDDAT ;WHAT REGISTER SHOULD BE
042614 005037 001124 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
042620 023737 001124 001126 BEQ 68$ ;BR IF OK
042626 001403 EMT 24
042630 104024 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
042632 005137 001244 68$: NOP
042636 000240 MOVB PORTA, RPCS2(R0) ;SELECT PORT A
042640 113760 001224 000010 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042646 013737 001224 001234 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042654 005037 001244 MOV RPER1(R0), $BDDAT ;GET CONTENTS OF RPER1
042660 016037 000014 001126 MOV #RPER1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042666 012737 000014 001122 ADD R0, $BDADR ;ADD RH11 BASE ADDRESS
042674 060037 001122 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
042700 012737 177777 001124 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
042706 023737 001124 001126 BEQ 70$ ;BR IF OK
042714 001403 EMT 10
042716 1C4010 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
042720 005137 001244 70$: NOP
042724 000240
  
```

;CLEAR THE ERRORS THROUGH PORT A

```

042726 012760 000011 000000 MOV #11, RPCS1(R0) ;ISSUE A DRIVE CLEAR
  
```

```

;RELEASE THE DRIVE FROM PORT A

042734 113760 001224 000010      MOVB   PORTA,RPCS2(R0)  ;SELECT PORT A
042742 013737 001224 001234      MOV    PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042750 012760 000013 000000      MOV    #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

042756 005037 001250              CLR    RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
042762 012737 000012 001122      MOV    #RPDS1,$BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
042770 060037 001122              ADD    R0,$BDADR      ;ADD THE I/O BASE ADDRESS
042774 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
043002 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
043010 016037 000012 001170      MOV    RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
043016 013737 001170 001164      MOV    $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
043024 042737 100100 001164      BIC    #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043032 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
043040 016037 000012 001172      MOV    RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
043046 013737 001172 001166      MOV    $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
043054 042737 100100 001166      BIC    #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043062 023737 001164 001166      CMP    $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
043070 001006              BNE    72$            ;BR IF NOT
043072 005737 001164              TST    $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
043076 001045              BNE    74$            ;BR IF NOT
043100 104046              EMT    46
043102 000137 043266              JMP    76$            ;BYPASS THE REST OF THE CHECKS
043106 013737 001170 001126 72$:  MOV    $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
043114 013737 001226 001234      MOV    PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043122 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
043130 005737 001164              TST    $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
043134 001414              SEQ    73$            ;BR IF ZERO
043136 013737 001224 001234      MOV    PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043144 013737 001172 001126      MOV    $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
043152 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
043160 005737 001166              TST    $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
043164 001012              BNE    74$            ;BR IF NOT
043166 012737 177777 001250 73$:  MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
043174 012760 000011 000000      MOV    #11,RPCS1(R0) ;CLEAR THE DRIVE
043202 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
043210 104026              EMT    26
043212 013737 001170 001126 74$:  MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
043220 013737 001224 001234      MOV    PORTA,PTNBR    ;CHANGE PORT NUMBER
043226 023737 001124 001170      CMP    $GDDAT,$TMP2   ;ALL BITS OK ?
043234 001401              BEQ    75$            ;BR IF OK FROM PORT A.
043236 104007              EMT    7
043240 013737 001172 001126 75$:  MOV    $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
043246 013737 001226 001234      MOV    PORTB,PTNBR    ;CHANGE PORT NUMBER
043254 023737 001124 001172      CMP    $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
043262 001401              BEQ    76$            ;BR IF OK
043264 104007              EMT    7
043266 000240              76$:  NOP
043270 000004              2$:  SCOPE                ;LOOP ?
  
```

394
395

```

*****
;*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.

 TST34:

```

043272      005737  001274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
043272      001406      BEQ      2$          ;BR IF NOT
043300      100002      BPL      1$          ;BR IF JUST ENTERED TEST
043302      000137  003010      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
043306      012737  177777  001274  1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
043314      112737  000034  001102  2$:      MOV      #34,$TSTNM ;TEST NUMBER
043322      012737  043344  001106      MOV      #TEST34,$LPADR ;LOAD LOOP ON TEST ADDRESS
043330      012737  043344  001110      MOV      #TEST34,$LPERR ;LOAD LOOP ON ERROR ADDRESS
043336      012737  007640  001176      MOV      #4000,$TIMES ;DO 4000. ITERATIONS
396 043344  012706  001100  TEST34: MOV      #STACK,$P ;LOAD THE STACK POINTER
397
;CLEAR ATTENTION BITS FOR BOTH PORTS
043350      113760  001224  000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT #A
043356      005060  000012      CLR      RPDS1(R0) ;SEIZE THE DRIVE
043362      012760  000011  000000      MOV      #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
043370      012760  000013  000000      MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
043376      113760  001226  000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT #B
043404      005060  000012      CLR      RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
043410      012760  000011  000000      MOV      #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
043416      012760  000013  000000      MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
;SEIZE THE DRIVE THROUGH PORT B
043424      113760  001226  000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT B
043432      013737  001226  001236      MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
043440      005060  000012      CLR      RPDS1(R0) ;WRITE RPDS1
043444      013737  001224  001240      MOV      PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
;FORCE AN ERROR
043452      012760  177777  000014      MOV      #-1,RPER1(R0) ;SET ERROR BITS
043460      012760  000013  000000      MOV      #13,RPCS1(R0) ;ISSUE A RELEASE COMMAND
043466      005037  001244      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
043472      016037  000000  001126      MOV      RPCS1(R0),$BDDAT ;GET CONTENTS OF RPCS1
043500      012737  000000  001122      MOV      #RPCS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043506      060037  001122      ADD      R0,$BDADR ;ADD RH11 BASE ADDRESS
043512      012737  004012  001124      MOV      #4012,$GDDAT ;WHAT REGISTER SHOULD BE
043520      013737  001126  001164      MOV      $BDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
043526      042737  173765  001164      BIC      #^C4012,$TMPO ;SAVE SPECIFIED BITS
    
```

```

043534 023737 001124 001164      CMP      $GDDAT,$STMP0      ;COMPARE THE BITS
043542 001414                      BEQ      66$                ;BR IF OK
043544 013737 001126 001174      MOV      $BDDAT,$STMP4     ;COPY 'BAD DATA'
043552 042737 004012 001174      BIC      #4012,$STMP4      ;CLEAR THE MASKED BITS
043560 053737 001174 001124      BIS      $STMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
043566 104025                      EMT      25
043570 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
043574 000240                      NOP
043576 005737 001244      66$:  TST      CKERR              ;DID 'GO' BIT RESET ?
043602 001002                      BNE      +6                ;BR IF NOT
043604 000137 043644                      JMP      1$                ;'GO' BIT RESET
043610 012760 000040 000010      MOV      #CLR,RPCS2(R0)    ;INIT THE RH11
043616 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B
043624 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043632 012760 000013 000000      MOV      #13,RPCS1(R0)    ;RELEASE THE DRIVE THROUGH PORT B
043640 000137 044360                      JMP      2$                ;BYPASS THE REST OF THE TEST
  
```

;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B

```

043644 113760 001224 000010      1$:  MOVB     PORTA,RPCS2(R0)   ;SELECT PORT A
043644 013737 001224 001234      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043660 005037 001244                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
043664 016037 000012 001126      MOV      RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
043672 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043700 060037 001122                      ADD      R0,$BDADR         ;ADD RH11 BASE ADDRESS
043704 005037 001124                      CLR      $GDDAT            ;WHAT REGISTER SHOULD BE
043710 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
043716 001403                      BEQ      68$                ;BR IF OK
043720 104024                      EMT      24
043722 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
043726 000240                      68$:  NOP
043730 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B
043736 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043744 005037 001244                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
043750 016037 000014 001126      MOV      RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
043756 012737 000014 001122      MOV      #RPER1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043764 060037 001122                      ADD      R0,$BDADR         ;ADD RH11 BASE ADDRESS
043770 012737 177777 001124      MOV      #177777,$GDDAT   ;WHAT REGISTER SHOULD BE
043776 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
044004 001403                      BEQ      70$                ;BR IF OK
044006 104010                      EMT      10
044010 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
044014 000240                      70$:  NOP
  
```

;CLEAR THE ERRORS THROUGH PORT B

```

044016 012760 000011 000000      MOV      #11,RPCS1(R0)    ;ISSUE A DRIVE CLEAR
  
```

;RELEASE THE DRIVE FROM PORT B

```

044024 113760 001226 000010      MOVB     PORTB,RPCS2(R0)   ;SELECT PORT B
044032 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044040 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

044046 005037 001250          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
044052 012737 000012 001122  MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
044060 060037 001122          ADD      R0,$BDADR   ;ADD THE I/O BASE ADDRESS
044064 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
044072 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
044100 016037 000012 001170  MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044106 013737 001170 001164  MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
044114 042737 100100 001164  BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044122 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
044130 016037 000012 001172  MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
044136 013737 001172 001166  MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
044144 042737 100100 001166  BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044152 023737 001164 001166  CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
044160 001006          BNE      72$          ;BR IF NOT
044162 005737 001164          TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
044166 001045          BNE      74$          ;BR IF NOT
044170 104046          EMT      46
044172 000137 044356          JMP      76$          ;BYPASS THE REST OF THE CHECKS
044176 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
044204 013737 001226 001234  MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
044212 113760 001226 000010  MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
044220 005737 001164          TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
044224 001414          BEQ      73$          ;BR IF ZERO
044226 013737 001224 001234  MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
044234 013737 001172 001126  MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
044242 113760 001224 000010  MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
044250 005737 001166          TST      $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
044254 001012          BNE      74$          ;BR IF NOT
044256 012737 177777 001250 73$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
044264 012760 000011 000000  MOV      #11,RPCS1(R0) ;CLEAR THE DRIVE
044272 012760 000013 000000  MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
044300 104026          EMT      26
044302 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
044310 013737 001224 001234  MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
044316 023737 001124 001170  CMP      $GDDAT,$TMP2  ;ALL BITS OK ?
044324 001401          BEQ      75$          ;BR IF OK FROM PORT A.
044326 104007          EMT      7
044330 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
044336 013737 001226 001234  MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
044344 023737 001124 001172  CMP      $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.
044352 001401          BEQ      76$          ;BR IF OK
044354 104007          EMT      7
044356 000240          76$:  NOP
044360 000004          2$:  SCOPE          ;LOOP ?
    
```

398
414
415

```

*****
*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.
:*

```

```

416 044362 005737 001274
451 044362 001406
044366 100002
044370 000137 003010
044372 012737 177777 001274 1$:
044376 112737 000035 001102 2$:
044404 012737 044434 001106
044412 012737 044434 001110
044420 012737 000004 001176
044426 012706 001100
TEST35:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      2$          ;BR IF NOT
      BPL      1$          ;BR IF JUST ENTERED TEST
      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
      MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
      MOV      #35,$STNM    ;TEST NUMBER
      MOV      #TEST35,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST35,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #4,$TIMES    ;DO 4 ITERATIONS
      MOV      #STACK,SP   ;LOAD THE STACK POINTER

      ;CLEAR ATTENTION BITS FOR BOTH PORTS
      MOV      PORTA,RPDS2(R0) ;SELECT PORT #A
      CLR      RPDS1(R0)      ;SEIZE THE DRIVE
      MOV      #11,RPDS1(R0)  ;ISSUE DRIVE CLEAR
      MOV      #13,RPDS1(R0)  ;RELEASE THE DRIVE
      MOV      PORTB,RPDS2(R0) ;SELECT PORT #B
      CLR      RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
      MOV      #11,RPDS1(R0)  ;ISSUE DRIVE CLEAR
      MOV      #13,RPDS1(R0)  ;RELEASE THE DRIVE

      ;SEIZE THE DRIVE THROUGH PORT A
      MOV      PORTA,RPDS2(R0) ;SELECT PORT A
      MOV      PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
      CLR      RPDS1(R0)      ;WRITE RPDS1
      MOV      PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS

      ;WAIT 500 MS

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

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

      ;RETRIGGER THE TIMEOUT ONE-SHOT
      TST      RPDS1(R0)     ;RETRIGGER THE ONE-SHOT
      MOV      PORTB,RPDS2(R0) ;SELECT PORT B
      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

044614 005760 000012      2$:  TST      RPDS1(R0)      ;WAIT FOR TIMEOUT
044620 001004              BNE      3$           ;BR IF TIMEOUT OCCURRED
044622 005737 001254      TST      WATCH        ;WATCH EQUAL TO ZERO ?
044626 001372              BNE      2$           ;BR IF NOT
044630 104036              EMT      36
044632 013737 001252 001272 3$:  MOV      TIME,TIMES   ;SAVE THE ELAPSED TIME VALUE

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

044640 005037 001250      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
044644 012737 000012 001122  MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
044652 060037 001122      ADD      R0,$BDADR    ;ADD THE I/O BASE ADDRESS
044656 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
044664 113760 001224 000010  MOVB     PORTA,RPCS2(R0) ;SELECT PORT A.
044672 016037 000012 001170  MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044700 013737 001170 001164  MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
044706 042737 100100 001164  BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044714 113760 001226 000010  MOVB     PORTB,RPCS2(R0) ;SELECT PORT B.
044722 016037 000012 001172  MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
044730 013737 001172 001166  MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
044736 042737 100100 001166  BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044744 023737 001164 001166  CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
044752 001006              BNE      66$          ;BR IF NOT
044754 005737 001164      TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
044760 001045              BNE      68$          ;BR IF NOT
044762 104046              EMT      46
044764 000137 045150      JMP      70$          ;BYPASS THE REST OF THE CHECKS
044770 013737 001170 001126 66$:  MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
044776 013737 001226 001234  MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045004 113760 001226 000010  MOVB     PORTB,RPCS2(R0) ;SELECT PORT B.
045012 005737 001164      TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
045016 001414              BEQ      67$          ;BR IF ZERO
045020 013737 001224 001234  MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045026 013737 001172 001126  MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
045034 113760 001224 000010  MOVB     PORTA,RPCS2(R0) ;SELECT PORT A.
045042 005737 001166      TST      $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
045046 001012              BNE      68$          ;BR IF NOT
045050 012737 177777 001250 67$:  MOV      #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
045056 012760 000011 000000  MOV      #11,RPCS1(R0) ;CLEAR THE DRIVE
045064 012760 000013 000000  MOV      #13,RPCS1(R0) ;RELEASE THE DRIVE
045072 104022              EMT      22
045074 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
045102 013737 001224 001234  MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
045110 023737 001124 001170  CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
045116 001401              BEQ      69$          ;BR IF OK FROM PORT A.
045120 104007              EMT      7
045122 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
045130 013737 001226 001234  MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
045136 023737 001124 001172  CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
045144 001401              BEQ      70$          ;BR IF OK
045146 104007              EMT      7
045150 000240              70$:  NOP

                                ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

045152 023737 001272 001260  CMP      TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
    
```

```

045160 003004          BGT      4$          :BR IF GREATER
045162 023737 001272 001262  CMP      TIMES,TIMEAM :MEASURED TIME LESS THAN -25% TOLERANCE
045170 002001          BGE      .+4          :BR IF NOT
045172          4$:          EMT      25
045172 104025          SCOPE          :LOOP ?
045174 000004

```

467
468

```

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

```

```

045176          TST36:  TST      KYBCTL          :PERFORMING ONLY SINGLE TESTS ?
045176 005737 001274  BEQ      2$          :BR IF NOT
045202 001406          BPL      1$          :BR IF JUST ENTERED TEST
045204 100002          JMP      EXEC          :RETURN & GET NEXT TEST NUMBER
045206 000137 003010  1$:      MOV      #-1,KYBCTL :SET SINGLE TEST INDICATOR
045212 012737 177777 001274 2$:      MOV      #36,$STNM  :TEST NUMBER
045220 112737 000036 001102  MOV      #TEST36,$LPADR :LOAD LOOP ON TEST ADDRESS
045226 012737 045250 001106  MOV      #TEST36,$LPERR :LOAD LOOP ON ERROR ADDRESS
045234 012737 045250 001110  MOV      #4,$TIMES      :DO 4 ITERATIONS
045242 012737 000004 001176  TEST36: MOV      #STACK,SP  :LOAD THE STACK POINTER
045250 012706 001100

```

469
470

```

: CLEAR ATTENTION BITS FOR BOTH PORTS
MOV      PORTA,RPDS2(R0) :SELECT PORT #A
CLR      RPDS1(R0)       :SEIZE THE DRIVE
MOV      #11,RPDS1(R0)   :ISSUE DRIVE CLEAR
MOV      #13,RPDS1(R0)   :RELEASE THE DRIVE
MOV      PORTB,RPDS2(R0) :SELECT PORT #B
CLR      RPDS1(R0)       :SEIZE THE DRIVE THROUGH PORT 'B'
MOV      #11,RPDS1(R0)   :ISSUE DRIVE CLEAR
MOV      #13,RPDS1(R0)   :RELEASE THE DRIVE

```

```

: SEIZE THE DRIVE THROUGH PORT B
MOV      PORTB,RPDS2(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
045356 005037 001252          CLR     TIME           ;CLEAR THE ELAPSED TIME COUNTER
045362 012737 000764 001254  MOV     #500.,WATCH    ;SET WATCH TO 500 MS
045370 005737 001254 1$:    TST     WATCH          ;WATCH EQUAL TO ZERO
045374 001375          BNE     1$             ;BR IF NOT

;START THE TIMER
045376 005037 001252          CLR     TIME           ;CLEAR THE ELAPSED TIME COUNTER
045402 012737 003720 001254  MOV     #2000.,WATCH   ;SET WATCH TO 2000 MS

;RETRIGGER THE TIMEOUT ONE-SHOT
045410 005760 000012          TST     RPDS1(R0)      ;RETRIGGER THE ONE-SHOT
045414 113760 001224 000010  MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
045422 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045430 005760 000012 2$:    TST     RPDS1(R0)      ;WAIT FOR TIMEOUT
045434 001004          BNE     3$             ;BR IF TIMEOUT OCCURRED
045436 005737 001254          TST     WATCH          ;WATCH EQUAL TO ZERO ?
045442 001372          BNE     2$             ;BR IF NOT
045444 104036          EMT     36
045446 013737 001252 001272 3$:  MOV     TIME,TIMES     ;SAVE THE ELAPSED TIME VALUE

;VERIFY THAT THE DRIVE IS IN NEUTRAL
045454 005037 001250          CLR     RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
045460 012737 000012 001122  MOV     #RPDS1,$BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
045466 060037 001122          ADD     R0,$BDADR     ;ADD THE I/O BASE ADDRESS
045472 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
045500 113760 001224 000010  MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
045506 016037 000012 001170  MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
045514 013737 001170 001164  MOV     $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
045522 042737 100100 001164  BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045530 113760 001226 000010  MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
045536 016037 000012 001172  MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
045544 013737 001172 001166  MOV     $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
045552 042737 100100 001166  BIC     #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045560 023737 001164 001166  CMP     $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
045566 001006          BNE     66$           ;BR IF NOT
045570 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
045574 001045          BNE     68$           ;BR IF NOT
045576 104046          EMT     46
045600 000137 045764          JMP     70$           ;BYPASS THE REST OF THE CHECKS
045604 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
045612 013737 001226 001234  MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045620 113760 001226 000010  MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
045626 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
045632 001414          BEQ     67$           ;BR IF ZERO
045634 013737 001224 001234  MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045642 013737 001172 001126  MOV     $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
045650 113760 001224 000010  MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
045656 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
045662 001012          BNE     68$           ;BR IF NOT
045664 012737 177777 001250 67$:  MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
045672 012760 000011 000000  MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE

```

```

045700 012760 000013 000000      MOV    #13,RPCS1(R0)    ;RELEASE THE DRIVE
045706 104022                      EMT    22
045710 013737 001170 001126 68$:  MOV    $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
045716 013737 001224 001234      MOV    PORTA,PTNBR     ;CHANGE PORT NUMBER
045724 023737 001124 001170      CMP    $GDDAT,$TMP2    ;ALL BITS OK ?
045732 001401                      BEQ    69$              ;BR IF OK FROM PORT A.
045734 104007                      EMT    7
045736 013737 001172 001126 69$:  MOV    $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
045744 013737 001226 001234      MOV    PORTB,PTNBR     ;CHANGE PORT NUMBER
045752 023737 001124 001172      CMP    $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
045760 001401                      BEQ    70$              ;BR IF OK
045762 104007                      EMT    7
045764 000240                      70$:  NOP
  
```

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

045766 023737 001272 001266      CMP    TIMES,TIMEBP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
045774 003004                      BGT    4$              ;BR IF GREATER
045776 023737 001272 001270      CMP    TIMES,TIMEBM    ;MEASURED TIME LESS THAN -25% TOLERANCE
046004 002001                      BGE    .+4             ;BR IF NOT
046006                                4$:  EMT    25
046006 104025                      SCOPE
046010 000004                                ;LOOP ?
  
```

471
472
489
490

```

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

```

046012                                TST37:
046012 005737 001274      TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
046016 001406                      BEQ    2$              ;BR IF NOT
046020 100002                      BPL    1$              ;BR IF JUST ENTERED TEST
046022 000137 003010      JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
046026 012737 177777 001274 1$:  MOV    #-1,KYBCiL      ;SET SINGLE TEST INDICATOR
046034 112737 000037 001102 2$:  MOV    #37,$STNM      ;TEST NUMBER
046042 012737 046064 001106      MOV    #TEST37,$LPADR ;LOAD LOOP ON TEST ADDRESS
046050 012737 046064 001110      MOV    #TEST37,$LPERR ;LOAD LOOP ON ERROR ADDRESS
046056 012737 000004 001176      MOV    #4,$TIMES      ;DO 4 ITERATIONS
491 046064 012706 001100      TEST37: MOV    #STACK,$P    ;LOAD THE STACK POINTER
518
  
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

046070 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT #A
046076 005060 000012              CLR    RPDS1(R0)      ;SEIZE THE DRIVE
046102 012760 000011 000000      MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
046110 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
046116 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT #B
046124 005060 000012              CLR    RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
046130 012760 000011 000000      MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
046136 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
046144 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
046152 013737 001224 001234      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046160 013737 001224 001236      MOV    PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
    
```

;DO A RECALIBRATE THROUGH PORT A

```

046166 012760 000007 000000      MOV    #7,RPCS1(R0)  ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
    
```

;WAIT FOR DRIVE TO FINISH

```

046174 032760 000200 000012      BIT    #DRY,RPDS1(R0) ;WAIT FOR DRIVE TO FINISH
046202 001774              BEQ    .-6            ;BR IF NOT FINISHED
    
```

;CONFIRM THAT ATTENTION IS SET FOR PORT A

```

046204 005037 001244              CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
046210 016037 000012 001126      MOV    RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
046216 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
046224 060037 001122              ADD    R0,$BDADR       ;ADD RH11 BASE ADDRESS
046230 012737 100000 001124      MOV    #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
046236 013737 001126 001164      MOV    $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
046244 042737 077777 001164      BIC    #^CATA,$STMP0   ;SAVE SPECIFIED BITS
046252 023737 001124 001164      CMP    $GDDAT,$STMP0   ;COMPARE THE BITS
046260 001414              BEQ    64$            ;BR IF OK
046262 013737 001126 001174      MOV    $BDDAT,$STMP4   ;COPY 'BAD DATA'
046270 042737 100000 001174      BIC    #ATA,$STMP4     ;CLEAR THE MASKED BITS
046276 053737 001174 001124      BIS    $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
046304 104032              EMT    32
046306 005137 001244              COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
046312 000240 64$: NOP
046314 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
046322 013737 001226 001234      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

```

046330 005037 001244              CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
046334 016037 000012 001126      MOV    RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
046342 012737 000012 001122      MOV    #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
046350 060037 001122              ADD    R0,$BDADR       ;ADD RH11 BASE ADDRESS
046354 005037 001124              CLR    $GDDAT         ;WHAT REGISTER SHOULD BE
046360 013737 001126 001164      MOV    $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
046366 042737 077777 001164      BIC    #^CATA,$STMP0   ;SAVE SPECIFIED BITS
046374 023737 001124 001164      CMP    $GDDAT,$STMP0   ;COMPARE THE BITS
046402 001414              BEQ    66$            ;BR IF OK
046404 013737 001126 001174      MOV    $BDDAT,$STMP4   ;COPY 'BAD DATA'
046412 042737 100000 001174      BIC    #ATA,$STMP4     ;CLEAR THE MASKED BITS
046420 053737 001174 001124      BIS    $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
046426 104032              EMT    32
046430 005137 001244              COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
    
```

046434 000240 66\$: NOP

;RELEASE THE DRIVE FROM PORT A

046436 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
046444 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046452 012760 000013 000000 MOV #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

046460 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
046464 012737 000012 001122 MOV #RPDS1,\$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
046472 060037 001122 ADD R0,\$BDDADR ;ADD THE I/O BASE ADDRESS
046476 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
046504 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
046512 016037 000012 001170 MOV RPDS1(R0),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
046520 013737 001170 001164 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
046526 042737 100100 001164 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046534 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
046542 016037 000012 001172 MOV RPDS1(R0),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046550 013737 001172 001166 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
046556 042737 100100 001166 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046564 023737 001164 001166 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046572 001006 BNE 68\$;BR IF NOT
046574 005737 001164 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046600 001045 BNE 70\$;BR IF NOT
046602 104046 EMT 46
046604 000137 046770 JMP 72\$;BYPASS THE REST OF THE CHECKS
046610 013737 001170 001126 68\$: MOV \$TMP2,\$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046616 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046624 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B.
046632 005737 001164 TST \$TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
046636 001414 BEQ 69\$;BR IF ZERO
046640 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046646 013737 001172 001126 MOV \$TMP3,\$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
046654 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A.
046662 005737 001166 TST \$TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
046666 001012 BNE 70\$;BR IF NOT
046670 012737 177777 001250 69\$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
046676 012760 000011 000000 MOV #11,RPCS1(R0) ;CLEAR THE DRIVE
046704 012760 000013 000000 MOV #13,RPCS1(R0) ;RELEASE THE DRIVE
046712 104026 EMT 26
046714 013737 001170 001126 70\$: MOV \$TMP2,\$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
046722 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
046730 023737 001124 001170 CMP \$GDDAT,\$TMP2 ;ALL BITS OK ?
046736 001401 BEQ 71\$;BR IF OK FROM PORT A.
046740 104007 EMT 7
046742 013737 001172 001126 71\$: MOV \$TMP3,\$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
046750 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
046756 023737 001124 001172 CMP \$GDDAT,\$TMP3 ;SEE IF READ OK FROM PORT B.
046764 001401 BEQ 72\$;BR IF OK
046766 104007 EMT 7
046770 000240 72\$: NOP
046772 000004 SCOPE ;LOOP ?

534
535

;*****

```

: *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.
: *
: *****

```

TST40: *****

```

046774
046774 005737 001274
047000 001406
047002 100002
047004 000137 003010
047010 012737 177777 001274
047016 112737 000040 001102
047024 012737 047046 001106
047032 012737 047046 001110
047040 012737 000004 001176
536 047046 012706 001100
537

```

```

TST40: 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      #40,$STNM  ;TEST NUMBER
        MOV      #TEST40,$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
        MOV      PORTA,$RPCS2(R0) ;SELECT PORT #A
        CLR      $RPDS1(R0)      ;SEIZE THE DRIVE
        MOV      #11,$RPCS1(R0)  ;ISSUE DRIVE CLEAR
        MOV      #13,$RPCS1(R0)  ;RELEASE THE DRIVE
        MOV      PORTB,$RPCS2(R0) ;SELECT PORT #B
        CLR      $RPDS1(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
        MOV      #11,$RPCS1(R0)  ;ISSUE DRIVE CLEAR
        MOV      #13,$RPCS1(R0)  ;RELEASE THE DRIVE
        MOV      PORTB,$RPCS2(R0) ;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

```

047150 012760 000007 000000      MOV      #7,$RPCS1(R0) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B

```

;WAIT FOR DRIVE TO FINISH

```

047156 032760 000200 000012      BIT      #DRY,$RPDS1(R0) ;WAIT FOR DRIVE TO FINISH
047164 001774                      BEQ      -6              ;BR IF NOT FINISHED

```

;CONFIRM THAT ATTENTION IS SET FOR PORT B

```

047166 005037 001244                      CLR      $CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
047172 016037 000012 001126      MOV      $RPDS1(R0),$BDDAT ;GET CONTENTS OF $RPDS1
047200 012737 000012 001122      MOV      #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
047206 060037 001122                      ADD      R0,$BDADR       ;ADD RH11 BASE ADDRESS
047212 012737 100000 001124      MOV      #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE

```



```

047220 013737 001126 001164      MOV      $BDDAT,$STMP0      ;MOVE REGISTER CONTENTS TO '$STMP0'
047226 042737 077777 001164      BIC      #^CATA,$STMP0     ;SAVE SPECIFIED BITS
047234 023737 001124 001164      CMP      $GDDAT,$STMP0     ;COMPARE THE BITS
047242 001414                      BEQ      64$                ;BR IF OK
047244 013737 001126 001174      MOV      $BDDAT,$STMP4     ;COPY 'BAD DATA'
047252 042737 100000 001174      BIC      #ATA,$STMP4       ;CLEAR THE MASKED BITS
047260 053737 001174 001124      BIS      $STMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
047266 104032                      EMT
047270 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
047274 000240      64$:      NOP
047276 113760 001224 000010      MOV      PORTA,RPCS2(R0)   ;SELECT PORT A
047304 013737 001224 001234      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

047312 005037 001244                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
047316 016037 000012 001126      MOV      RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
047324 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
047332 060037 001122                      ADD      R0,$BDADR         ;ADD RH11 BASE ADDRESS
047336 005037 001124                      CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
047342 013737 001126 001164      MOV      $BDDAT,$STMP0     ;MOVE REGISTER CONTENTS TO '$STMP0'
047350 042737 077777 001164      BIC      #^CATA,$STMP0     ;SAVE SPECIFIED BITS
047356 023737 001124 001164      CMP      $GDDAT,$STMP0     ;COMPARE THE BITS
047364 001414                      BEQ      66$                ;BR IF OK
047366 013737 001126 001174      MOV      $BDDAT,$STMP4     ;COPY 'BAD DATA'
047374 042737 100000 001174      BIC      #ATA,$STMP4       ;CLEAR THE MASKED BITS
047402 053737 001174 001124      BIS      $STMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
047410 104032                      EMT
047412 005137 001244                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
047416 000240      66$:      NOP

;RELEASE THE DRIVE FROM PORT B

047420 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B
047426 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047434 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

047442 005037 001250                      CLR      RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
047446 012737 000012 001122      MOV      #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
047454 060037 001122                      ADD      R0,$BDADR         ;ADD THE I/O BASE ADDRESS
047460 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
047466 113760 001224 000010      MOV      PORTA,RPCS2(R0)   ;SELECT PORT A.
047474 016037 000012 001170      MOV      RPDS1(R0),$STMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
047502 013737 001170 001164      MOV      $STMP2,$STMP0     ;COPY IT INTO '$STMP0'
047510 042737 100100 001164      BIC      #ATA!VV,$STMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047516 113760 001226 000010      MOV      PORTB,RPCS2(R0)   ;SELECT PORT B.
047524 016037 000012 001172      MOV      RPDS1(R0),$STMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
047532 013737 001172 001166      MOV      $STMP3,$STMP1     ;COPY IT INTO '$STMP1'
047540 042737 100100 001166      BIC      #ATA!VV,$STMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047546 023737 001164 001166      CMP      $STMP0,$STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
047554 001006                      BNE      68$                ;BR IF NOT
047556 005737 001164                      TST      $STMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
047562 001045                      BNE      70$                ;BR IF NOT
047564 104046                      EMT      46

```

```

047566 000137 047752          JMP      72$          ;BYPASS THE REST OF THE CHECKS
047572 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047600 013737 001226 001234      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047606 113760 001226 000010      MOVVB  PORTB,RPCS2(R0) ;SELECT PORT B.
047614 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
047620 001414          BEQ     69$          ;BR IF ZERO
047622 013737 001224 001234      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047630 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
047636 113760 001224 000010      MOVVB  PORTA,RPCS2(R0) ;SELECT PORT A.
047644 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
047650 001012          BNE     70$          ;BR IF NOT
047652 012737 177777 001250 69$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
047660 012760 000011 000000      MOV     #11,RPCS1(R0) ;CLEAR THE DRIVE
047666 012760 000013 000000      MOV     #13,RPCS1(R0) ;RELEASE THE DRIVE
047674 104026          EMT     26
047676 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
047704 013737 001224 001234      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
047712 023737 001124 001170      CMP     $GDDAT,$TMP2 ;ALL BITS OK ?
047720 001401          BEQ     71$          ;BR IF OK FROM PORT A.
047722 104007          EMT     7
047724 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
047732 013737 001226 001234      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
047740 023737 001124 001172      CMP     $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
047746 001401          BEQ     72$          ;BR IF OK
047750 104007          EMT     7
047752 000240          72$:  NOP
047754 000004          SCOPE          ;LOOP ?
  
```

538
 53
 564

```

*****
*TEST 41      TEST PORT INTERACTION FROM PORT 'A'
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
*  B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
*
*  C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT
*     'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
*
*  E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT
*     PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS
*     SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
*     SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
*
*  G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
*     RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
  
```

```

047756 005737 001274
047756 001406
047762
  
```

```

TST41:  TST     KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
        BEQ     2$      ;BR IF NOT
  
```

```

047764 100002          BPL      1$          ;BR IF JUST ENTERED TEST
047766 000137 003010   JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
047772 012737 177777 001274 1$:   MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
050000 112737 000041 001102 2$:   MOVVB   #41,$TSTNM  ;TEST NUMBER
050006 012737 050030 001106   MOV      #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
050014 012737 050030 001110   MOV      #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
050022 012737 007640 001176   MOV      #4000,$TIMES  ;DO 4000. ITERATIONS
565 050030 012706 001100   TEST41: MOV     #STACK,$SP ;LOAD THE STACK POINTER
600

;CLEAR ATTENTION BITS FOR BOTH PORTS

050034 113760 001224 000010   MOVVB   PORTA,$RPCS2(R0) ;SELECT PORT #A
050042 005060 000012          CLR      RPDS1(R0)       ;SEIZE THE DRIVE
050046 012760 000011 000000   MOV      #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
050054 012760 000013 000000   MOV      #13,$RPCS1(R0) ;RELEASE THE DRIVE
050062 113760 001226 000010   MOVVB   PORTB,$RPCS2(R0) ;SELECT PORT #B
050070 005060 000012          CLR      RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
050074 012760 000011 000000   MOV      #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
050102 012760 000013 000000   MOV      #13,$RPCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

050110 113760 001224 000010   MOVVB   PORTA,$RPCS2(R0) ;SELECT PORT A
050116 013737 001224 001236   MOV      PORTA,$SEIZPT ;STORE SEIZING PORT'S ADDRESS
050124 005060 000012          CLR      RPDS1(R0)       ;WRITE RPDS1
050130 013737 001226 001240   MOV      PORTB,$OPPRT   ;'OPPOSITE' PORT ADDRESS
050136 012760 177777 000014   MOV      #-1,$RPER1(R0) ;LOAD 1'S INTO RPER1 THROUGH PORT A
050144 012760 177777 000040   MOV      #-1,$RPER2(R0) ;LOAD 1'S INTO RPER2 THROUGH PORT A
050152 012760 177777 000042   MOV      #-1,$RPER3(R0) ;LOAD 1'S INTO RPER3 THROUGH PORT A
050160 113760 001226 000010   MOVVB   PORTB,$RPCS2(R0) ;SELECT PORT B
050166 013737 001226 001234   MOV      PORTB,$PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050174 004737 051032          JSR      PC,$TST41B     ;CHECK THE REGISTERS THROUGH PORT B
050200 113760 001224 000010   MOVVB   PORTA,$RPCS2(R0) ;SELECT PORT A
050206 013737 001224 001234   MOV      PORTA,$PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050214 005060 000042          CLR      RPER3(R0)       ;CLEAR RPER3 ON PORT A
050220 005060 000040          CLR      RPER2(R0)       ;CLEAR RPER2 ON PORT A
050224 005060 000014          CLR      RPER1(R0)       ;CLEAR RPER1 ON PORT A
050230 013760 001232 000016   MOV      ASR1,$RPAS(R0) ;CLEAR THE ATTENTION BIT FOR PORT A
050236 113760 001226 000010   MOVVB   PORTB,$RPCS2(R0) ;SELECT PORT B
050244 013737 001226 001234   MOV      PORTB,$PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050252 012760 177777 000014   MOV      #-1,$RPER1(R0) ;LOAD 1'S INTO RPER1 THROUGH PORT B
050260 012760 177777 000040   MOV      #-1,$RPER2(R0) ;LOAD 1'S INTO RPER2 THROUGH PORT B
050266 012760 177777 000042   MOV      #-1,$RPER3(R0) ;LOAD 1'S INTO RPER3 THROUGH PORT B
050274 113760 001224 000010   MOVVB   PORTA,$RPCS2(R0) ;SELECT PORT A
050302 013737 001224 001234   MOV      PORTA,$PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050310 004737 051032          JSR      PC,$TST41B     ;CHECK THE REGISTERS THROUGH PORT A

;RELEASE THE DRIVE FROM PORT A

050314 113760 001224 000010   MOVVB   PORTA,$RPCS2(R0) ;SELECT PORT A
050322 013737 001224 001234   MOV      PORTA,$PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050330 012760 000013 000000   MOV      #13,$RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

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

050336 005037 001250          CLR      RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
050342 012737 111700 001124   MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GGDAT ;COMPARISON CONSTANT
    
```

```

050350 012737 000012 001122      MOV      #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
050356 060037 001122              ADD      R0,$BDADR    ;REGISTER BASE ADDRESS FOR TYPEOUT
050362 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B
050370 013737 001226 001234      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050376 016037 000012 001164      MOV      RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
050404 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A
050412 013737 001224 001234      MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050420 016037 000012 001126      MOV      RPDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT A
050426 001404              BEQ      66$          ;BR IF STATUS FROM PORT A ZERO
050430 005737 001164              TST      $TMP0        ;IS STATUS FROM PORT B ZERO ?
050434 001401              BEQ      66$          ;BR IF ZERO
050436 104031              EMT      31
050440 013737 001164 001126 66$:   MOV      $TMP0,$BDDAT   ;CHECK STATUS FROM PORT B
050446 013737 001226 001234      MOV      PORTB,PTNBR   ;CHANGE PORT ADDRESS FOR TYPEOUT
050454 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
050462 001401              BEQ      67$          ;BR IF OK
050464 104027              EMT      27
050466 000240 67$:   NOP

;RELEASE THE DRIVE FROM PORT B

050470 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B
050476 013737 001226 001234      MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050504 012760 000013 000000      MOV      #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

050512 005037 001250              CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
050516 012737 000012 001122      MOV      #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
050524 060037 001122              ADD      R0,$BDADR    ;ADD THE I/O BASE ADDRESS
050530 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
050536 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
050544 016037 000012 001170      MOV      RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
050552 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
050560 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050566 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
050574 016037 000012 001172      MOV      RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
050602 013737 001172 001166      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
050610 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050616 023737 001164 001166      CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
050624 001006              BNE      68$          ;BR IF NOT
050626 005737 001164              TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
050632 001045              BNE      70$          ;BR IF NOT
050634 104046              EMT      46
050636 000137 051022              JMP      72$          ;BYPASS THE REST OF THE CHECKS
050642 013737 001170 001126 68$:   MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
050650 013737 001226 001234      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050656 113760 001226 000010      MOVVB   PORTB,RPCS2(R0) ;SELECT PORT B.
050664 005737 001164              TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
050670 001414              BEQ      69$          ;BR IF ZERO
050672 013737 001224 001234      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050700 013737 001172 001126      MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
050706 113760 001224 000010      MOVVB   PORTA,RPCS2(R0) ;SELECT PORT A.
050714 005737 001166              TST      $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
050720 001012              BNE      70$          ;BR IF NOT
050722 012737 177777 001250 69$:   MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
050730 012760 000011 000000      MOV      #11,RPCS1(R0) ;CLEAR THE DRIVE

```

```

050736 012760 000013 000000      MOV      #13,RPCS1(R0)      ;RELEASE THE DRIVE
050744 104026                      EMT      26
050746 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
050754 013737 001224 001234      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
050762 023737 001124 001170      CMP      $GDDAT,$TMP2     ;ALL BITS OK ?
050770 001401                      BEQ      71$              ;BR IF OK FROM PORT A.
050772 104007                      EMT      7
050774 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
051002 013737 001226 001234      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
051010 023737 001124 001172      CMP      $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
051016 001401                      BEQ      72$              ;BR IF OK
051020 104007                      EMT      7
051022 000240                      NOP
051024 000004                      SCOPE
601 051026 000137 051364      JMP      TST42             ;LOOP ?
                                ;GO TO THE NEXT TEST
    
```

;CHECK THE REGISTERS ON THE SELECTED PORT

```

051032                                TST41B:
051032 005037 001244                      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
051036 016037 000014 001126      MOV      RPER1(R0),$BDDAT ;GET CONTENTS OF RPER1
051044 012737 000014 001122      MOV      #RPER1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051052 060037 001122                      ADD      R0,$BDADR        ;ADD RH11 BASE ADDRESS
051056 005037 001124                      CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
051062 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
051070 001403                      BEQ      64$              ;BR IF OK
051072 104006                      EMT      6
051074 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
051100 016037 000000 001126 64$:  MOV      RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHCS1
051106 012737 000000 001122      MOV      #RPCS1,$BDADR    ;FORM ADDRESS OF REGISTER
051114 060037 001122                      ADD      R0,$BDADR        ;ADDRESS BASE
051120 032737 020000 001126      BIT      #MCPE,$BDDAT     ;IS 'MCPE' SET ?
051126 001404                      BEQ      65$              ;BR IF NOT
051130 104011                      EMT      11
051132 012760 040000 000000      MOV      #TRE,RPCS1(R0)   ;CLEAR 'MCPE'
051140 000240                      NOP
051142 005037 001244                      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
051146 016037 000040 001126      MOV      RPER2(R0),$BDDAT ;GET CONTENTS OF RPER2
051154 012737 000040 001122      MOV      #RPER2,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051162 060037 001122                      ADD      R0,$BDADR        ;ADD RH11 BASE ADDRESS
051166 005037 001124                      CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
051172 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
051200 001403                      BEQ      66$              ;BR IF OK
051202 104006                      EMT      6
051204 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
051210 016037 000000 001126 66$:  MOV      RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHC1
051216 012737 000000 001122      MOV      #RPCS1,$BDADR    ;FORM ADDRESS OF REGISTER
051224 060037 001122                      ADD      R0,$BDADR        ;ADDRESS BASE
051230 032737 020000 001126      BIT      #MCPE,$BDDAT     ;IS 'MCPE' SET ?
051236 001404                      BEQ      67$              ;BR IF NOT
051240 104011                      EMT      11
051242 012760 040000 000000      MOV      #TRE,RPCS1(R0)   ;CLEAR 'MCPE'
051250 000240                      NOP
051252 005037 001244                      CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
051256 016037 000042 001126      MOV      RPER3(R0),$BDDAT ;GET CONTENTS OF RPER3
051264 012737 000042 001122      MOV      #RPER3,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051272 060037 001122                      ADD      R0,$BDADR        ;ADD RH11 BASE ADDRESS
    
```

```

051276 005037 001124          CLR    $GDDAT      ;WHAT REGISTER SHOULD BE
051302 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
051310 001403                BEQ    68$         ;BR IF OK
051312 104006                EMT
051314 005137 001244          COM    CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
051320 016037 000000 001126 68$: MOV    RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHCS1
051326 012737 000000 001122  MOV    #RPCS1,$BDADR ;FORM ADDRESS OF REGISTER
051334 060037 001122          ADD    R0,$BDADR   ;ADDRESS BASE
051340 032737 020000 001126  BIT    #MCPE,$BDDAT ;IS 'MCPE' SET ?
051346 001404                BEQ    69$         ;BR IF NOT
051350 104011                EMT
051352 012760 040000 000000 69$: MOV    #TRE,RPCS1(R0) ;CLEAR 'MCPE'
051360 000240                NOP
051362 000207                RTS     PC          ;RETURN
    
```

626
627

```

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

```

051364 005737 001274          TST42: TST    KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
051364 001406                BEQ    2$         ;BR IF NOT
051372 100002                BPL    1$         ;BR IF JUST ENTERED TEST
051374 000137 003010          JMP    EXEC       ;RETURN & GET NEXT TEST NUMBER
051400 012737 177777 001274 1$: MOV    #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
051406 112737 000042 001102 2$: MOV    #42,$STNM   ;TEST NUMBER
051414 012737 051436 001106  MOV    #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
051422 012737 051436 001110  MOV    #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
051430 012737 007640 001176  MOV    #4000,$TIMES  ;DO 4000. ITERATIONS
628 051436 012706 001100  TEST42: MOV    #STACK,SP ;LOAD THE STACK POINTER
629
;CLEAR ATTENTION BITS FOR BOTH PORTS
051442 113760 001224 000010  MOV    PORTA,RPCS2(R0) ;SELECT PORT #A
051450 005060 000012          CLR    RPDS1(R0)   ;SEIZE THE DRIVE
051454 012760 000011 000000  MOV    #11,RPCS1(R0) ;ISSUE DRIVE CLEAR
    
```

```

051462 012760 000013 000000      MOV      #13,RPCS1(R0)      ;RELEASE THE DRIVE
051470 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT #B
051476 005060 000012 000000      CLR      RPDS1(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
051502 012760 000011 000000      MOV      #11,RPCS1(R0)    ;ISSUE DRIVE CLEAR
051510 012760 000013 000000      MOV      #13,RPCS1(R0)    ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

051516 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT B
051524 013737 001226 001236      MOV      PORTB,SEIZPT      ;STORE SEIZING PORT'S ADDRESS
051532 005060 000012 000000      CLR      RPDS1(R0)         ;WRITE RPDS1
051536 013737 001224 001240      MOV      PORTA,OPPRT       ;'OPPOSITE' PORT ADDRESS
051544 012760 177777 000014      MOV      #-1,RPER1(R0)     ;LOAD 1'S INTO RPER1 THROUGH PORT B
051552 012760 177777 000040      MOV      #-1,RPER2(R0)     ;LOAD 1'S INTO RPER2 THROUGH PORT B
051560 012760 177777 000042      MOV      #-1,RPER3(R0)     ;LOAD 1'S INTO RPER3 THROUGH PORT B
051566 113760 001224 000010      MOVB     PORTA,RPCS2(R0)    ;SELECT PORT A
051574 013737 001224 001234      MOV      PORTA,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051602 004737 052440 000000      JSR      PC,TST42B         ;CHECK THE REGISTERS THROUGH PORT A
051606 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT B
051614 013737 001226 001234      MOV      PORTB,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051622 005060 000042 000000      CLR      RPER3(R0)         ;CLEAR RPER3 ON PORT B
051626 005060 000040 000000      CLR      RPER2(R0)         ;CLEAR RPER2 ON PORT B
051632 005060 000014 000000      CLR      RPER1(R0)         ;CLEAR RPER1 ON PORT B
051636 013760 001232 000016      MOV      ASR1,RPAS(R0)     ;CLEAR THE ATTENTION BIT FOR PORT B
051644 113760 001224 000010      MOVB     PORTA,RPCS2(R0)    ;SELECT PORT A
051652 013737 001224 001234      MOV      PORTA,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051660 012760 177777 000014      MOV      #-1,RPER1(R0)     ;LOAD 1'S INTO RPER1 THROUGH PORT A
051666 012760 177777 000040      MOV      #-1,RPER2(R0)     ;LOAD 1'S INTO RPER2 THROUGH PORT A
051674 012760 177777 000042      MOV      #-1,RPER3(R0)     ;LOAD 1'S INTO RPER3 THROUGH PORT A
051702 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT B
051710 013737 001226 001234      MOV      PORTB,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051716 004737 052440 000000      JSR      PC,TST42B         ;CHECK THE REGISTERS THROUGH PORT B

                                ;RELEASE THE DRIVE FROM PORT B

051722 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT B
051730 013737 001226 001234      MOV      PORTB,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051736 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

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

051744 005037 001250 000000      CLR      RELERR           ;CLEAR 'RELEASE ERROR' INDICATOR
051750 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
051756 012737 000012 001122      MOV      #RPDS1,$BDADR     ;REGISTER ADDRESS INCREMENT
051764 060037 001122 000000      ADD      R0,$BDADR         ;REGISTER BASE ADDRESS FOR TYPEOUT
051770 113760 001224 000010      MOVB     PORTA,RPCS2(R0)    ;SELECT PORT A
051776 013737 001224 001234      MOV      PORTA,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052004 016037 000012 001164      MOV      RPDS1(R0),$TMP0    ;READ STATUS REGISTER FROM PORT A
052012 113760 001226 000010      MOVB     PORTB,RPCS2(R0)    ;SELECT PORT B
052020 013737 001226 001234      MOV      PORTB,PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052026 016037 000012 001126      MOV      RPDS1(R0),$BDDAT   ;DRIVE STATUS FROM PORT B
052034 001404 000000 000000      BEQ      66$              ;BR IF STATUS FROM PORT B ZERO
052036 005737 001164 000000      TST      $TMP0            ;IS STATUS FROM PORT A ZERO ?
052042 001401 000000 000000      BEQ      66$              ;BR IF ZERO
052044 104031 000000 000000      EMT      31
052046 013737 001164 001126      MOV      $TMP0,$BDDAT      ;CHECK STATUS FROM PORT A
052054 013737 001224 001234      MOV      PORTA,PTNBR       ;CHANGE PORT ADDRESS FOR TYPEOUT
    
```


630 052434 000137 052772 JMP TST43 ;GO TO THE NEXT TEST
 ;CHECK THE REGISTERS ON THE SELECTED PORT

```

052440                                TST42B:
052440 005037 001244                    CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
052444 016037 000014 001126            MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
052452 012737 000014 001122            MOV #RPER1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052460 060037 001122                    ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
052464 005037 001124                    CLR $GDDAT ;WHAT REGISTER SHOULD BE
052470 023737 001124 001126            CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
052476 001403 001124 001126            BEQ 64$ ;BR IF OK
052500 104006                            EMT 6
052502 005137 001244                    COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
052506 016037 000000 001126 64$:      MOV RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
052514 012737 000000 001122            MOV #RPCS1, $BDADR ;FORM ADDRESS OF REGISTER
052522 060037 001122                    ADD RO, $BDADR ;ADDRESS BASE
052526 032737 020000 001126            BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
052534 001404 001124 001126            BEQ 65$ ;BR IF NOT
052536 104011                            EMT 11
052540 012760 040000 000000            MOV #TRE, RPCS1(RO) ;CLEAR 'MCPE'
052546 000240                            NOP
052550 005037 001244                    CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
052554 016037 000040 001126            MOV RPER2(RO), $BDDAT ;GET CONTENTS OF RPER2
052562 012737 000040 001122            MOV #RPER2, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052570 060037 001122                    ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
052574 005037 001124                    CLR $GDDAT ;WHAT REGISTER SHOULD BE
052600 023737 001124 001126            CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
052606 001403 001124 001126            BEQ 66$ ;BR IF OK
052610 104006                            EMT 6
052612 005137 001244                    COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
052616 016037 000000 001126 66$:      MOV RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
052624 012737 000000 001122            MOV #RPCS1, $BDADR ;FORM ADDRESS OF REGISTER
052632 060037 001122                    ADD RO, $BDADR ;ADDRESS BASE
052636 032737 020000 001126            BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
052644 001404 001124 001126            BEQ 67$ ;BR IF NOT
052646 104011                            EMT 11
052650 012760 040000 000000            MOV #TRE, RPCS1(RO) ;CLEAR 'MCPE'
052656 000240                            NOP
052660 005037 001244                    CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
052664 016037 000042 001126            MOV RPER3(RO), $BDDAT ;GET CONTENTS OF RPER3
052672 012737 000042 001122            MOV #RPER3, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052700 060037 001122                    ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
052704 005037 001124                    CLR $GDDAT ;WHAT REGISTER SHOULD BE
052710 023737 001124 001126            CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
052716 001403 001124 001126            BEQ 68$ ;BR IF OK
052720 104006                            EMT 6
052722 005137 001244                    COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
052726 016037 000000 001126 68$:      MOV RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
052734 012737 000000 001122            MOV #RPCS1, $BDADR ;FORM ADDRESS OF REGISTER
052742 060037 001122                    ADD RO, $BDADR ;ADDRESS BASE
052746 032737 020000 001126            BIT #MCPE, $BDDAT ;IS 'MCPE' SET ?
052754 001404 001124 001126            BEQ 69$ ;BR IF NOT
052756 104011                            EMT 11
052760 012760 040000 000000            MOV #TRE, RPCS1(RO) ;CLEAR 'MCPE'
052766 000240                            NOP
052770 000207                            RTS PC ;RETURN
  
```

631
 644
 645

```

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

052772
 052772 005737 001274
 052776 001406
 053000 100002
 053002 000137 C03010
 053006 012737 177777 001274
 053014 112737 000043 001102
 053022 012737 053044 001106
 053030 012737 053044 001110
 053036 012737 000031 001176
 646 053044 012706 001100
 679

```

TST43:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOV      #43,$TSTNM ;TEST NUMBER
          MOV      #TEST43,$LPADR ;LOAD LOOP ON TEST ADDRESS
          MOV      #TEST43,$LPERR ;LOAD LOOP ON ERROR ADDRESS
          MOV      #25,$TIMES ;DO 25. ITERATIONS
TEST43:  MOV      #STACK,$SP ;LOAD THE STACK POINTER
    
```

:CLEAR ATTENTION BITS FOR BOTH PORTS

053050 113760 001224 000010
 053056 005060 000012
 053062 012760 000011 000000
 053070 012760 000013 000000
 053076 113760 001226 000010
 053104 005060 000012
 053110 012760 000011 000000
 053116 012760 000013 000000
 053124 113760 001224 000010
 053132 012760 177777 000014
 053140 005060 000014
 053144 113760 001226 000010
 053152 005760 000012
 053156 001775
 053160 012737 000016 001122
 053166 060037 001122
 053172 013737 001232 001124
 053200 013737 001232 001166
 053206 005137 001166
 053212 012737 053246 001110
 053220 113760 001226 000010
 053226 013737 001226 001234
 053234 013737 001226 001236
 053242 005060 000012
 053246 016037 000016 001126
 053254 013737 001126 001164
 053262 043737 001166 001164
 053270 023737 001124 001164

```

MOV      PORTA,$RPCS2(R0) ;SELECT PORT #A
CLR      RPDS1(R0)        ;SEIZE THE DRIVE
MOV      #11,$RPCS1(R0)   ;ISSUE DRIVE CLEAR
MOV      #13,$RPCS1(R0)   ;RELEASE THE DRIVE
MOV      PORTB,$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
MOV      PORTA,$RPCS2(R0) ;SELECT PORT A
MOV      #-1,$RPER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT A
CLR      RPER1(R0)        ;CLEAR THE ERRORS
1$:      MOV      PORTB,$RPCS2(R0) ;SELECT PORT B
          TST      RPDS1(R0)        ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
          BEQ      1$                ;BR IF STILL SEIZED BY PORT A
          MOV      #RPAS,$BDADR      ;FORM ADDRESS OF ATTN REG IF ERROR
          ADD      R0,$BDADR          ;ADD THE ADDRESS BASE
          MOV      ASR1,$GDDAT       ;GOOD DATA FOR ERROR MESSAGE
          MOV      ASR1,$STMP1       ;MAKE DATA COMPARE MASK
          COM      $STMP1            ;COMPLEMENT IT
          MOV      #2$,$LPERR        ;LOAD LOOP ON ERROR ADDRESS
          MOV      PORTB,$RPCS2(R0) ;SELECT PORT B
          MOV      PORTB,$PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
          MOV      PORTB,$SEIZPT     ;'SEIZED' PORT ADDRESS
          CLR      RPDS1(R0)        ;SEIZE THE DRIVE THROUGH PORT B
          MOV      RPAS(R0),$BDAT    ;GET THE CONTENTS OF THE ATTENTION REG
          MOV      $BDAT,$STMP0     ;PUT CONTENTS INTO WORKING LOCATION
          BIC      $STMP1,$STMP0    ;CLEAR OTHER BITS
          CMP      $GDDAT,$STMP0    ;SEE IF ATTN BIT FOR DRIVE SET
    
```

```

053276 001401          BEQ      3$          :BR IF SET
053300 104053          EMT      53
053302          3$:

;RELEASE THE DRIVE FROM PORT B

053302 113760 001226 000010  MOVB   PORTB,RPCS2(R0)  :SELECT PORT B
053310 013737 001226 001234  MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
053316 012760 000013 000000  MOV    #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

053324 005037 001250          CLR    RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
053330 012737 000012 001122  MOV    #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
053336 060037 001122          ADD    R0,$BDADR      ;ADD THE I/O BASE ADDRESS
053342 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
053350 113760 001224 000010  MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
053356 016037 000012 001170  MOV    RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
053364 013737 001170 001164  MOV    $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
053372 042737 100100 001164  BIC    #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
053400 113760 001226 000010  MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
053406 016037 000012 001172  MOV    RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
053414 013737 001172 001166  MOV    $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
053422 042737 100100 001166  BIC    #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
053430 023737 001164 001166  CMP    $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
053436 001006          BNE    64$          ;BR IF NOT
053440 005737 001164          TST    $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
053444 001045          BNE    66$          ;BR IF NOT
053446 104046          EMT      46
053450 000137 053650          JMP    68$          ;BYPASS THE REST OF THE CHECKS
053454 013737 001170 001126 64$: MOV    $TMP2,$BDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
053462 013737 001226 001234  MOV    PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053470 113760 001226 000010  MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
053476 005737 001164          TST    $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
053502 001414          BEQ    65$          ;BR IF ZERO
053504 013737 001224 001234  MOV    PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053512 013737 001172 001126  MOV    $TMP3,$BDAT     ;'BAD DATA' FOR ERROR TYPE OUT
053520 113760 001224 000010  MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
053526 005737 001166          TST    $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
053532 001012          BNE    66$          ;BR IF NOT
053534 012737 177777 001250 65$: MOV    #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
053542 012760 000011 000000  MOV    #11,RPCS1(R0)   ;CLEAR THE DRIVE
053550 012760 000013 000000  MOV    #13,RPCS1(R0)   ;RELEASE THE DRIVE
053556 104026          EMT      26
053560 013737 001170 001126 66$: MOV    $TMP2,$BDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
053566 013737 001224 001234  MOV    PORTA,PTNBR     ;CHANGE PORT NUMBER
053574 042737 100000 001170  BIC    #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
053602 023737 001124 001170  CMP    $GDDAT,$TMP2   ;ALL BITS OK ?
053610 001401          BEQ    67$          ;BR IF OK FROM PORT A.
053612 104007          EMT      7
053614 013737 001172 001126 67$: MOV    $TMP3,$BDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
053622 013737 001226 001234  MOV    PORTB,PTNBR     ;CHANGE PORT NUMBER
053630 042737 100000 001172  BIC    #ATA,$TMP3     ;DON'T CHECK THE ATTN BIT
053636 023737 001124 001172  CMP    $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
053644 001401          BEQ    68$          ;BR IF OK
053646 104007          EMT      7
053650 000240          68$: NOP
    
```

692
693

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.
*
:*****
  
```

053654
053654
053660
053662
053664
053670
053676
053704
053712
053720
694 053726
695

005737 001274
001406
100002
000137 003010
012737 177777 001274
112737 000044 001102
012737 053726 001106
012737 053726 001110
012737 000031 001176
012706 001100

```

TST44:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #44,$TSTNM ;TEST NUMBER
MOV #TEST44,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST44,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #25,$TIMES ;DO 25. ITERATIONS
TEST44: MOV #STACK,SP ;LOAD THE STACK POINTER
  
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

053732 113760 001224 000010
053740 005060 000012
053744 012760 000011 000000
053752 012760 000013 000000
053760 113760 001226 000010
053766 005060 000012
053772 012760 000011 000000
054000 012760 000013 000000
054006 113760 001226 000010
054014 012760 177777 000014
054022 005060 000014
054026 113760 001224 000010
054034 005760 000012
054040 001775
054042 012737 000016 001122
054050 060037 001122
054054 013737 001232 001124
054062 013737 001232 001166
054070 005137 001166
054074 012737 054130 001110
054102 113760 001224 000010
054110 013737 001224 001234
054116 013737 001224 001236
054124 005060 000012
054130 016037 000016 001126
054136 013737 001126 001164
054144 043737 001166 001164
054152 023737 001124 001164

```

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
MOVb PORTB,RPCS2(R0) ;SELECT PORT B
MOV #-1,RPER1(R0) ;SET ERRORS TO FORCE A'TN BIT ON PORT B
CLR RPER1(R0) ;CLEAR THE ERRORS
1$: MOVb PORTA,RPCS2(R0) ;SELECT PORT A
TST RPDS1(R0) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
BEQ 1$ ;BR IF STILL SEIZED BY PORT B
MOV #RPAS,$BDADR ;FORM ADDRESS OF ATTN REG IF ERROR
ADD R0,$BDADR ;ADD THE ADDRESS BASE
MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
COM $TMP1 ;COMPLEMENT IT
MOV #2$,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOVb PORTA,RPCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
CLR RPDS1(R0) ;SEIZE THE DRIVE THROUGH PORT A
2$: MOV RPAS(R0),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
MOV $BDDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
  
```

```

054160 001401          BEQ    3$          ;BR IF SET
054162 104053          EMT    53
054164          3$:

;RELEASE THE DRIVE FROM PORT A

054164 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT A
054172 013737 001224 001234  MOV   PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054200 012760 000013 000000  MOV   #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

054206 005037 001250          CLR   RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
054212 012737 000012 001122  MOV   #RPDS1,$BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
054220 060037 001122          ADD   R0,$BDADR      ;ADD THE I/O BASE ADDRESS
054224 012737 011700 001124  MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
054232 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT A.
054240 016037 000012 001170  MOV   RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054246 013737 001170 001164  MOV   $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
054254 042737 100100 001164  BIC   #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054262 113760 001226 000010  MOVB  PORTB,RPCS2(R0) ;SELECT PORT B.
054270 016037 000012 001172  MOV   RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054276 013737 001172 001166  MOV   $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
054304 042737 100100 001166  BIC   #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054312 023737 001164 001166  CMP   $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054320 001006          BNE   64$          ;BR IF NOT
054322 005737 001164          TST   $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054326 001045          BNE   66$          ;BR IF NOT
054330 104046          EMT    46
054332 000137 054532          JMP   68$          ;BYPASS THE REST OF THE CHECKS
054336 013737 001170 001126 64$: MOV   $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054344 013737 001226 001234  MOV   PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054352 113760 001226 000010  MOVB  PORTB,RPCS2(R0) ;SELECT PORT B.
054360 005737 001164          TST   $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
054364 001414          BEQ   65$          ;BR IF ZERO
054366 013737 001224 001234  MOV   PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054374 013737 001172 001126  MOV   $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
054402 113760 001224 000010  MOVB  PORTA,RPCS2(R0) ;SELECT PORT A.
054410 005737 001166          TST   $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
054414 001012          BNE   66$          ;BR IF NOT
054416 012737 177777 001250 65$: MOV   #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
054424 012760 000011 000000  MOV   #11,RPCS1(R0)  ;CLEAR THE DRIVE
054432 012760 000013 000000  MOV   #13,RPCS1(R0)  ;RELEASE THE DRIVE
054440 104026          EMT    26
054442 013737 001170 001126 66$: MOV   $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
054450 013737 001224 001234  MOV   PORTA,PTNBR    ;CHANGE PORT NUMBER
054456 042737 100000 001170  BIC   #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
054464 023737 001124 001170  CMP   $GDDAT,$TMP2   ;ALL BITS OK ?
054472 001401          BEQ   67$          ;BR IF OK FROM PORT A.
054474 104007          EMT    7
054476 013737 001172 001126 67$: MOV   $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
054504 013737 001226 001234  MOV   PORTB,PTNBR    ;CHANGE PORT NUMBER
054512 042737 100000 001172  BIC   #ATA,$TMP3     ;DON'T CHECK THE ATTN BIT
054520 023737 001124 001172  CMP   $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
054526 001401          BEQ   68$          ;BR IF OK
054530 104007          EMT    7
054532 000240          68$: NOP
    
```

696 054534 000004 SCOPE :LOOP ?
 697 054536 000137 056460 JMP \$EOP :GO TO END OF TEST
 698
 699
 700
 701
 722
 723

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

 *TEST 45 TEST NO TIMEOUT THROUGH PORT 'A'

*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
 * SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

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

 TST45:

054542	005737	001274			TST	KYBCTL	:PERFORMING ONLY SINGLE TESTS ?
054542	001406				BEQ	2\$:BR IF NOT
054550	100002				BPL	1\$:BR IF JUST ENTERED TEST
054552	000137	003010			JMP	EXEC	:RETURN & GET NEXT TEST NUMBER
054556	012737	177777	001274	1\$:	MOV	#-1,KYBCTL	:SET SINGLE TEST INDICATOR
054564	112737	000045	001102	2\$:	MOVB	#45,\$STSTM	:TEST NUMBER
054572	012737	054614	001106		MOV	#TEST45,\$LPADR	:LOAD LOOP ON TEST ADDRESS
054600	012737	054614	001110		MOV	#TEST45,\$LPERR	:LOAD LOOP ON ERROR ADDRESS
054606	012737	000004	001176		MOV	#4,\$TIMES	:DO 4 ITERATIONS
724 054614	012706	001100			TEST45: MOV	#STACK,SP	:LOAD THE STACK POINTER
765							

:CLEAR ATTENTION BITS FOR BOTH PORTS

054620	113760	001224	000010		MOVB	PORTA,RPCS2(R0)	:SELECT PORT #A
054626	005060	000012			CLR	RPDS1(R0)	:SEIZE THE DRIVE
054632	012760	000011	000000		MOV	#11,RPCS1(R0)	:ISSUE DRIVE CLEAR
054640	012760	000013	000000		MOV	#13,RPCS1(R0)	:RELEASE THE DRIVE
054646	113760	001226	000010		MOVB	PORTB,RPCS2(R0)	:SELECT PORT #B
054654	005060	000012			CLR	RPDS1(R0)	:SEIZE THE DRIVE THROUGH PORT 'B'
054660	012760	000011	000000		MOV	#11,RPCS1(R0)	:ISSUE DRIVE CLEAR
054666	012760	000013	000000		MOV	#13,RPCS1(R0)	:RELEASE THE DRIVE

:SEIZE THE DRIVE THROUGH PORT B

054674	113760	001226	000010		MOVB	PORTB,RPCS2(R0)	:SELECT PORT B
054702	013737	001226	001236		MOV	PORTB,SEIZPT	:STORE SEIZING PORT'S ADDRESS

```

054710 005060 000012          CLR    RPDS1(R0)      ;WRITE RPDS1
054714 013737 001224 001240    MOV    PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
054722 113760 001224 000010    MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
054730 013737 001224 001234    MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET REQUEST THROUGH PORT A

054736 005060 000012          CLR    RPDS1(R0)      ;SET REQUEST FOR PORT A
054742 113760 001226 000010    MOVB   PORTB,RPCS2(R0) ;SELECT PORT B
054750 013737 001226 001234    MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;RELEASE THE DRIVE THROUGH PORT B

054756 012760 000013 000000    MOV    #13,RPCS1(R0) ;RELEASE DRIVE THROUGH PORT B

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

054764 013737 001260 001254    MOV    TIMEAP,WATCH  ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A

054772 005037 001244          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
054776 016037 000012 001126    MOV    RPDS1(R0),%BDDAT ;GET CONTENTS OF RPDS1
055004 012737 000012 001122    MOV    #RPDS1,%BADDR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055012 060037 001122          ADD    R0,%BADDR     ;ADD RH11 BASE ADDRESS
055016 005037 001124          CLR    %GDDAT        ;WHAT REGISTER SHOULD BE
055022 023737 001124 001126    CMP    %GDDAT,%BDDAT  ;IS THE REGISTER OK ?
055030 001403          BEQ    66$          ;BR IF OK
055032 104031          EMT    31
055034 005137 001244          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
055040 000240          66$: NOP
055042 005737 001244          TST    CKERR         ;REGISTER OK ?
055046 001402          BEQ    .+6          ;BR IF OK
055050 000137 055506          JMP    1$           ;BYPASS REST OF TEST IF NOT
055054 005737 001254          TST    WATCH        ;WATCH EQUAL ZERO ?
055060 001375          BNE    .-4          ;BR IF NOT

;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT

055062 013737 001224 001234    MOV    PORTA,PTNBR   ;PORT NUMBER FOR TYPEOUT
055070 005037 001244          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
055074 016037 000012 001126    MOV    RPDS1(R0),%BDDAT ;GET CONTENTS OF RPDS1
055102 012737 000012 001122    MOV    #RPDS1,%BADDR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055110 060037 001122          ADD    R0,%BADDR     ;ADD RH11 BASE ADDRESS
055114 005037 001124          CLR    %GDDAT        ;WHAT REGISTER SHOULD BE
055120 023737 001124 001126    CMP    %GDDAT,%BDDAT  ;IS THE REGISTER OK ?
055126 001403          BEQ    68$          ;BR IF OK
055130 104035          EMT    35
055132 005137 001244          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
055136 000240          68$: NOP
055140 005737 001244          TST    CKERR         ;REGISTER OK ?
055144 001402          BEQ    .+6          ;BR IF OK
055146 000137 055506          JMP    1$           ;BYPASS REST OF TEST IF NOT

;RELEASE THE DRIVE FROM PORT A
    
```

```

055152 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A
055160 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
055166 012760 000013 000000      MOV    #13,RPCS1(R0) ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

055174 005037 001250                CLR    RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
055200 012737 000012 001122      MOV    #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
055206 060037 001122                ADD    R0,$BDADR ;ADD THE I/O BASE ADDRESS
055212 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
055220 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
055226 016037 000012 001170      MOV    RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
055234 013737 001170 001164      MOV    $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
055242 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055250 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
055256 016037 000012 001172      MOV    RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
055264 013737 001172 001166      MOV    $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
055272 042737 100100 001166      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055300 023737 001164 001166      CMP    $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
055306 001006                BNE    70$ ;BR IF NOT
055310 005737 001164                TST    $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
055314 001045                BNE    72$ ;BR IF NOT
055316 104046                EMT    46
055320 000137 055504                JMP    74$ ;BYPASS THE REST OF THE CHECKS
055324 013737 001170 001126 70$:  MOV    $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
055332 013737 001226 001234      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
055340 113760 001226 000010      MOVB   PORTB,RPCS2(R0) ;SELECT PORT B.
055346 005737 001164                TST    $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
055352 001414                BEQ    71$ ;BR IF ZERO
055354 013737 001224 001234      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
055362 013737 001172 001126      MOV    $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
055370 113760 001224 000010      MOVB   PORTA,RPCS2(R0) ;SELECT PORT A.
055376 005737 001166                TST    $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
055402 001012                BNE    72$ ;BR IF NOT
055404 012737 177777 001250 71$:  MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
055412 012760 000011 000000      MOV    #11,RPCS1(R0) ;CLEAR THE DRIVE
055420 012760 000013 000000      MOV    #13,RPCS1(R0) ;RELEASE THE DRIVE
055426 104026                EMT    26
055430 013737 001170 001126 72$:  MOV    $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
055436 013737 001224 001234      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
055444 023737 001124 001170      CMP    $GDDAT,$TMP2 ;ALL BITS OK ?
055452 001401                BEQ    73$ ;BR IF OK FROM PORT A.
055454 104007                EMT    7
055456 013737 001172 001126 73$:  MOV    $TMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
055464 013737 001226 001234      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
055472 023737 001124 001172      CMP    $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
055500 001401                BEQ    74$ ;BR IF OK
055502 104007                EMT    7
055504 000240                74$:  NOP

055506 000004                1$:  SCOPE ;LOOP ?

```

786
787

```

:*****
:*TEST 46 TEST NO TIMEOUT THROUGH PORT 'B'
:*
:*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
:* SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

```


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

TST46:

```

055510      055510 005737 001274
055514      055514 001406
055516      055516 100002
055520      055520 000137 003010
055524      055524 012737 177777 001274
055532      055532 112737 000046 001102
055540      055540 012737 055562 001106
055546      055546 012737 055562 001110
055554      055554 012737 000004 001176
788 055562 055562 012706 001100
789

TST46: TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
      BEQ      2$          ;BR IF NOT
      BPL      1$          ;BR IF JUST ENTERED TEST
      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:    MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
2$:    MOV      #46,$STNM   ;TEST NUMBER
      MOV      #TEST46,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV      #TEST46,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV      #4,$TIMES    ;DO 4 ITERATIONS
TEST46: MOV     #STACK,SP   ;LOAD THE STACK POINTER

      ;CLEAR ATTENTION BITS FOR BOTH PORTS
055566      055566 113760 001224 000010
055574      055574 005060 000012
055600      055600 012760 000011 000000
055606      055606 012760 000013 000000
055614      055614 113760 001226 000010
055622      055622 005060 000012
055626      055626 012760 000011 000000
055634      055634 012760 000013 000000
      MOV      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
      MOV      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 A
055642      055642 113760 001224 000010
055650      055650 013737 001224 001236
055656      055656 005060 000012
055662      055662 013737 001226 001240
055670      055670 113760 001226 000010
055676      055676 013737 001226 001234
      MOV      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
      MOV      PORTB,RPCS2(R0) ;SELECT PORT B
      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

      ;SET REQUEST THROUGH PORT B
055704      055704 005060 000012
055710      055710 113760 001224 000010
055716      055716 013737 001224 001234
      CLR      RPDS1(R0)      ;SET REQUEST FOR PORT B
      MOV      PORTA,RPCS2(R0) ;SELECT PORT A
      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

      ;RELEASE THE DRIVE THROUGH PORT A
    
```

```

055724 012760 000013 000000      MOV      #13,RPCS1(R0)      ;RELEASE DRIVE THROUGH PORT A
                                     ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)

055732 013737 001266 001254      MOV      TIMEBP,WATCH      ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
                                     ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

055740 005037 001244                CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
055744 016037 000012 001126      MOV      RPDS1(R0),%BDDAT  ;GET CONTENTS OF RPDS1
055752 012737 000012 001122      MOV      %RPDS1,%BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055760 060037 001122                ADD      R0,%BDADR        ;ADD RH11 BASE ADDRESS
055764 005037 001124                CLR      %GDDAT           ;WHAT REGISTER SHOULD BE
055770 023737 001124 001126      CMP      %GDDAT,%BDDAT    ;IS THE REGISTER OK ?
055776 001403                BEQ      66$              ;BR IF OK
056000 104031                EMT      31
056002 005137 001244                COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
056006 000240                66$:  NOP
056010 005737 001244                TST      CKERR            ;REGISTER OK ?
056014 001402                BEQ      +6              ;BR IF OK
056016 000137 056454                JMP      1$              ;BYPASS REST OF TEST IF NOT
056022 005737 001254                TST      WATCH           ;WATCH EQUAL ZERO ?
056026 001375                BNE      -4              ;BR IF NOT

                                     ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT

056030 013737 001226 001234      MOV      PORTB,PTNBR      ;PORT NUMBER FOR TYPEOUT
056036 005037 001244                CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
056042 016037 000012 001126      MOV      RPDS1(R0),%BDDAT  ;GET CONTENTS OF RPDS1
056050 012737 000012 001122      MOV      %RPDS1,%BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056056 060037 001122                ADD      R0,%BDADR        ;ADD RH11 BASE ADDRESS
056062 005037 001124                CLR      %GDDAT           ;WHAT REGISTER SHOULD BE
056066 023737 001124 001126      CMP      %GDDAT,%BDDAT    ;IS THE REGISTER OK ?
056074 001403                BEQ      68$              ;BR IF OK
056076 104035                EMT      35
056100 005137 001244                COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
056104 000240                68$:  NOP
056106 005737 001244                TST      CKERR            ;REGISTER OK ?
056112 001402                BEQ      +6              ;BR IF OK
056114 000137 056454                JMP      1$              ;BYPASS REST OF TEST IF NOT

                                     ;RELEASE THE DRIVE FROM PORT B

056120 113760 001226 000010      MOV      PORTB,RPCS2(R0)  ;SELECT PORT B
056126 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
056134 012760 000013 000000      MOV      #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL

056142 005037 001250                CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
056146 012737 000012 001122      MOV      %RPDS1,%BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
056154 060037 001122                ADD      R0,%BDADR        ;ADD THE I/O BASE ADDRESS
056160 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,%GDDAT ;COMPARISON CONSTANT
056166 113760 001224 000010      MOV      PORTA,RPCS2(R0)  ;SELECT PORT A.
056174 016037 000012 001170      MOV      RPDS1(R0),%TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
056202 013737 001170 001164      MOV      %TMP2,%TMP0      ;COPY IT INTO 'TMP0'
  
```

```

056210 042737 100100 001164      BIC      #ATA!VV,$TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
056216 113760 001226 000010      MOV      PORTB,RPCS2(R0)  ;SELECT PORT B.
056224 016037 000012 001172      MOV      RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
056232 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
056240 042737 100100 001166      BIC      #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
056246 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
056254 001006                      BNE      70$             ;BR IF NOT
056256 005737 001164                      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
056262 001045                      BNE      72$             ;BR IF NOT
056264 104046                      EMT      46
056266 000137 056452                      JMP      74$             ;BYPASS THE REST OF THE CHECKS
056272 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
056300 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056306 113760 001226 000010      MOV      PORTB,RPCS2(R0) ;SELECT PORT B.
056314 005737 001164                      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
056320 001414                      BEQ      71$             ;BR IF ZERO
056322 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056330 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
056336 113760 001224 000010      MOV      PORTA,RPCS2(R0) ;SELECT PORT A.
056344 005737 001166                      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
056350 001012                      BNE      72$             ;BR IF NOT
056352 012737 177777 001250 71$:  MOV      #-1,RELEERR     ;SET 'RELEASE ERROR' INDICATOR
056360 012760 000011 000000      MOV      #11,RPCS1(R0)   ;CLEAR THE DRIVE
056366 012760 000013 000000      MOV      #13,RPCS1(R0)   ;RELEASE THE DRIVE
056374 104026                      EMT      26
056376 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
056404 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
056412 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
056420 001401                      BEQ      73$             ;BR IF OK FROM PORT A.
056422 104007                      EMT      7
056424 013737 001172 001126 73$:  MOV      $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
056432 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
056440 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
056446 001401                      BEQ      74$             ;BR IF OK
056450 104007                      EMT      7
056452 000240 74$:  NOP

056454 000004 1$:  SCOPE                      ;LOOP ?

790
791      ;PUT NEWTEST HERE
794      ;*****
056456 000240      TST47: NOP
  
```

1

.SBTTL END OF PASS ROUTINE

```

:*****
:*INCREMENT THE PASS NUMBER ($PASS)
:*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
:*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY'
:*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
:*IF THERES A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO TST1AA
    
```

```

056460          056460 005737 001274
056464          056464 001402
056466          056466 000137 003010
056472          056472 005037 001102
056476          056476 005037 001176
056502          056502 005237 001100
056506          056506 042737 100000 001100
056514          056514 005327
056516          056516 000001
056520          056520 003066
056522          056522 012737
056524          056524 000001
056526          056526 056516
056530          056530 104401 056536
056534          056534 000407

056554          056554 013746 001100

056560          056560 104405
056562          056562 005737 001112
056566          056566 001431
056570          056570 104401 056576
056574          056574 000421

056640          056640 013746 001112

056644          056644 104405
056646          056646 005037 001112
056652          056652 104401 001207
056656          056656 013700 000042
056662          056662 001405
056664          056664 000005
056666          056666 004710
056670          056670 000240
056672          056672 000240
056674          056674 000240
056676          056676 000137
056700          056700 003304
056702          056702 377 377 000

$EOP:
TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
BEQ .+6 ;BR IF NOT
JMP EXEC ;RETURN TO KEYBOARD CONTROL
CLR $TSTNM ;ZERO THE TEST NUMBER
CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
INC $PASS ;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;LOOP?

$EOPCT: .WORD 1
BGT $DOAGN ;:YES
MOV (PC)+,@(PC)+ ;:RESTORE COUNTER

$ENDCT: .WORD 1
TYPE ,65$ ;:TYPE ASCIZ STRING
BR 64$ ;:GET OVER THE ASCIZ
::65$: .ASCIZ <12><15>/END PASS #/
64$:
MOV $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
;:TYPE PASS NUMBER
;:GO TYPE--DECIMAL ASCII WITH SIGN
TYPDS
TST $ERTTL ;:SEE IF ANY ERRORS THIS PASS
BEQ $GT42P ;:BR IF NO ERRORS TO REPORT
TYPE ,67$ ;:TYPE ASCIZ STRING
BR 66$ ;:GET OVER THE ASCIZ
::67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$
MOV $ERTTL,-(SP) ;:SAVE $ERTTL FOR TYPEOUT
;:TOTAL NUMBER OF ERRORS
;:GO TYPE--DECIMAL ASCII WITH SIGN
TYPDS
CLR $ERTTL ;:CLEAR ERROR TOTAL
,$CRLF ;:TYPE CARRIAGE RETURN, LINE FEED
$GT42P: MOV @#42,R0 ;:GET MONITOR ADDRESS
$GET42: BEQ $DOAGN ;:BRANCH IF NO MONITOR
RESET ;:CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;:GO TO MONITOR
NOP ;:SAVE ROOM
NOP ;:FOR
NOP ;:ACT11

$DOAGN:
JMP @(PC)+ ;:RETURN
$RTNAD: .WORD TST1AA
$ENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
.EVEN
    
```

.SBTTL CLOCK SUBROUTINES

```

2
3
4
5
6 056706 012737 056756 000004 CKCLK: MOV #CKCLK1,@#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
7 056714 005037 000006 CLR @#ERRVEC+2 ;NEW PSW
8 056720 005777 122266 TST @SLKCSR ;CHECK FOR KW11-P
9 056724 013701 001216 MOV $LPVEC,R1 ;KW11-P VECTOR ADDRESS
10 056730 012721 057040 MOV #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
11 056734 012711 000300 MOV #300,(R1) ;PSW - PRI 6
12 056740 012777 177777 122246 MOV #-1,@SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
13 056746 012777 000135 122236 MOV #135,@SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
14 056754 000425 BR CKCLK3
15 056756 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
16 056762 012737 057020 000004 MOV #CKCLK2,@#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
17 056770 005777 122224 TST @SLKS ;LOOK FOR KW11-L
18 056774 013701 001222 MOV $LLVEC,R1 ;KW11-L VECTOR ADDRESS
19 057000 012721 057040 MOV #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
20 057004 012711 000300 MOV #300,(R1) ;PSW - PRI 6
21 057010 012777 000100 122202 MOV #100,@SLKS ;SET KW11-L INTERRUPT
22 057016 000404 BR CKCLK3
23 057020 062706 000004 CKCLK2: ADD #4,SP ;RESTORE THE STACK POINTER
24 057024 062716 000002 ADD #2,(SP) ;INCREMENT RETURN, NO CLOCK
25 057030 012737 000006 000004 CKCLK3: MOV #6,@#ERRVEC ;RESTORE THE ERROR VECTOR
26 057036 000207 RTS PC
27
28 ;ROUTINE TO COUNT CLOCK TICKS
29
30 057040 062737 000021 001252 CLOCK: ADD #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
31 057046 005737 001254 TST WATCH ;IS WATCH ALREADY ZERO ?
32 057052 001406 BEQ 1$ ;BR IF IT IS
33 057054 152737 000021 001254 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
34 057062 100002 BPL 1$ ;BR IF NOT MINUS
35 057064 005037 001254 CLR WATCH ;CLEAR WATCH DOG COUNTER
36 057070 000002 1$: RTI ;RETURN
37
38 ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
39
40 057072 162706 000004 TOLER: SUB #4,SP ;SETUP STACK
41 057076 016616 000004 MOV 4(SP),(SP) ;SAVE STACK
42 057102 013546 MOV @R5+,-(SP) ;GET TIME VALUE
43 057104 011666 000004 MOV (SP),4(SP) ;MOVE TIME VALUE
44 057110 011666 000006 MOV (SP),6(SP) ;MOVE VALUE AGAIN
45 057114 006216 ASR (SP) ;DIVIDE BY 2
46 057116 006216 ASR (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
47 057120 061666 000004 ADD (SP),4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
48 057124 162666 000004 SUB (SP)+,4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
49 057130 000205 RTS R5 ;RETURN WITH TOLERANCES ON THE STACK
    
```

1

.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*CALL
*          SCOPE          ;;SCOPE=IOT
    
```

```

057132          $SCOPE:
057132 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
057134 032777 040000 121776 1$: BIT #BIT14,@SWR          ;;LOOP ON PRESENT TEST?
057142 001402          BEQ 9$          ;;NO IF SW14=0
057144 000137 057514          JMP $OVER          ;;JUMP OVER SCOPE ROUTINE
057150          9$:
          :#####START OF CODE FOR THE XOR TESTER#####
057150 000416          $XTSTR: BR 6$          ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
          :THIS INSTRUCTION TO A 'NOP' (NOP=240)
057152 013746 000004          MOV @ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
057156 012737 057176 000004          MOV #5$,@ERRVEC          ;;SET FOR TIMEOUT
057164 005737 177060          TST @#177060          ;;TIME OUT ON XOR?
057170 012637 000004          MOV (SP)+,@ERRVEC          ;;RESTORE THE ERROR VECTOR
057174 000534          BR $SVLAD          ;;GO TO THE NEXT TEST
057176 022626          5$: CMP (SP)+,(SP)+          ;;CLEAR THE STACK AFTER A TIME OUT
057200 012637 000004          MOV (SP)+,@ERRVEC          ;;RESTORE THE ERROR VECTOR
057204 000474          BR 7$          ;;LOOP ON THE PRESENT TEST
057206          6$:;#####END OF CODE FOR THE XOR TESTER#####
057206 105737 001103          2$: TSTB $ERFLG          ;;HAS AN ERROR OCCURRED?
057212 001502          BEQ 3$          ;;BR IF NO
057214 022737 177777 060054          CMP #-1,CPSAVE          ;;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED
057222 001455          BEQ 2003$          ;;KICK AROUND ROUTINE IF SO
057224 013746 000004          MOV ERRVEC,-(SP)          ;;SAVE CONTENTS OF ERROR VECTOR
057230 012737 057246 000004          MOV #2000$,ERRVEC          ;;SETUP 'TRAP' RETURN ADDRESS
057236 013737 177766 060054          MOV 177766,CPSAVE          ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
057244 000406          BR 2001$
057246 012737 177777 060054 2000$: MOV #-1,CPSAVE          ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
057254 012716 057262          MOV #2001$,(SP)          ;;SETUP RETURN ADDRESS
057260 000002          RTI
057262 012637 000004          2001$: MOV (SP)+,ERRVEC          ;;RESTORE CONTENTS OF ERROR VECTOR

057266 022737 177777 060054 2002$: CMP #-1,CPSAVE          ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
057274 001430          BEQ 2003$          ;;BRANCH IF SO
057276 032737 000001 060054          BIT #BIT00,CPSAVE          ;;SEE IF THE POWER MONITOR BIT IS ON
057304 001424          BEQ 2003$          ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
057306 042737 000001 177766          BIC #BIT00,177766          ;;CLEAR THE BIT FOUND TO BE SET
057314 013746 001140          MOV SWR,-(SP)          ;;SAVE SWR ADDRESS
057320 017646 000000          MOV @(SP),-(SP)          ;;SAVE SWR VALUE
057324 012737 000176 001140          MOV #176,SWR          ;;GET SOFTWARE SWR ADDRESS
057332 011677 121602          MOV (SP),@SWR          ;;GET CURRENT SWR VALUE
057336 042777 001000 121574          BIC #BIT09,@SWR          ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
057344 104177          EMT 177          ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
057346 012676 000000          MOV (SP)+,@(SP)          ;;RESTORE SWR TO ORIGINAL VALUE
057352 012637 001140          MOV (SP)+,SWR          ;;RESTORE SWR ADDRESS
    
```

```

057356          2003$:
057356 123737 001115 001103  CMPB  $ERMAX,$ERFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
057364 101015          BHI   3$                ;;BR IF NO
057366 032777 001000 121544  BIT   #BIT09,@SWR     ;;LOOP ON ERROR?
057374 001404          BEQ   4$                ;;BR IF NO
057376 013737 001110 001106  7$:  MOV   $LPERR,$LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
057404 000443          BR    $OVER
057406 105037 001103          4$:  CLRB  $ERFLG          ;;ZERO THE ERROR FLAG
057412 005037 001176          CLR   $TIMES         ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
057416 000415          BR    1$                ;;ESCAPE TO THE NEXT TEST
057420 032777 004000 121512  3$:  BIT   #BIT11,@SWR     ;;INHIBIT ITERATIONS?
057426 001011          BNE   1$                ;;BR IF YES
057430 005737 001100          TST   $PASS          ;;IF FIRST PASS OF PROGRAM
057434 001406          BEQ   1$                ;;      INHIBIT ITERATIONS
057436 005237 001104          INC   $ICNT          ;;INCREMENT ITERATION COUNT
057442 023737 001176 001104  CMP   $TIMES,$ICNT   ;;CHECK THE NUMBER OF ITERATIONS MADE
057450 002021          BGE   $OVER          ;;BR IF MORE ITERATION REQUIRED
057452 012737 000001 001104  1$:  MOV   #1,$ICNT       ;;REINITIALIZE THE ITERATION COUNTER
057460 013737 057530 001176  MOV   $MXCNT,$TIMES  ;;SET NUMBER OF ITERATIONS TO DO
057466 105237 001102          $SVLAD: INCB  $YSTNM   ;;COUNT TEST NUMBERS
057472 011637 001106          MOV   (SP),$LPADR   ;;SAVE SCOPE LOOP ADDRESS
057476 011637 001110          MOV   (SP),$LPERR   ;;SAVE ERROR LOOP ADDRESS
057502 005037 001200          CLR   $ESCAPE       ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
057506 112737 000001 001115  MOVB  #1,$ERMAX      ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
057514 013777 001102 121420 $OVER: MOV   $STNM,@DISPLAY  ;;DISPLAY TEST NUMBER
057522 013716 001106          MOV   $LPADR,(SP)   ;;FUDGE RETURN ADDRESS
057526 000002          RTI
057530 000004          $MXCNT: 4      ;;MAX. NUMBER OF ITERATIONS
    
```

.SBTTL ERROR HANDLER ROUTINE

```

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
*AND GO TO $ERRTYP ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1      HALT ON ERROR
*SW13=1      INHIBIT ERROR TYPEOUTS
*SW10=1      BELL ON ERROR
*CALL
*          ERROR      N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER
    
```

```

057532 105037 060056 $ERROR: CLRB      IBSAVE          ;;CLEAR THE ITEM BYTE SAVE LOCATION
057536 104407                CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
057540 113737 001102 001242      MOVB      $STNM,TSTNUM
057546 105237 001103      INCB      $ERFLG          ;;SET THE ERROR FLAG
057552 001775      BEQ      7$          ;;DON'T LET THE FLAG GO TO ZERO
057554 013777 001102 121360      MOV      $STNM,@DISPLAY  ;;DISPLAY TEST NUMBER AND ERROR FLAG
057552 032777 002000 121350      BIT      #BIT10,@SWR    ;;BELL ON ERROR?
057570 001402      BEQ      1$          ;;NO - SKIP
057572 104401 001202      TYPE     ,SBELL        ;;RING BELL
057576 005237 001112      INC      $ERTTL        ;;COUNT THE NUMBER OF ERRORS
057602 011637 001116      MOV      (SP),$ERRPC    ;;GET ADDRESS OF ERROR INSTRUCTION
057606 162737 000002 001116      SUB      #2,$ERRPC
057614 117737 121276 001114      MOV      @ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
057622 032777 001000 121310      BIT      #BIT09,@SWR    ;;SEE IF LOOP ON ERROR IS SET
057630 001060      BNE      1004$        ;;BRANCH AROUND ROUTINE IF SO
057632 122737 000177 001114      CMPB     #177,$ITEMB   ;;SEE IF THIS IS THE POWER FAIL CALL
057640 001454      BEQ      1004$        ;;BRANCH AROUND ROUTINE IF IT IS
057642 105737 060056      TSTB     IBSAVE        ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
057646 001047      BNE      1003$        ;;BRANCH IF SO
057650 022737 177777 060054      CMP      #-1,CPSAVE    ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
057656 001445      BEQ      1004$        ;;BRANCH IF SO
057660 013746 000004      MOV      ERRVEC,-(SP)   ;;SAVE CONTENTS OF ERROR VECTOR
057664 012737 057702 000004      MOV      #1000$,ERRVEC ;;SETUP 'TRAP' RETURN ADDRESS
057672 013737 177766 060054      MOV      177766,CPSAVE ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
057700 000406      BR       1001$
057702 012737 177777 060054 1000$: MOV      #-1,CPSAVE    ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
057710 012716 057716      MOV      #1001$, (SP)  ;;SETUP RETURN ADDRESS
057714 000002      RTI
057716 012637 000004      1001$: MOV      (SP)+,ERRVEC  ;;RESTORE CONTENTS OF ERROR VECTOR

057722 022737 177777 060054 1002$: CMP      #-1,CPSAVE    ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
057730 001420      BEQ      1004$        ;;BRANCH IF SO
057732 032737 000001 060054      BIT      #BIT00,CPSAVE  ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
057740 001414      BEQ      1004$        ;;BRANCH IF OK
057742 042737 000001 177766      BIC      #BIT00,177766  ;;CLEAR THE BIT FOUND SET
057750 113737 001114 060056      MOV      $ITEMB,IBSAVE  ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
057756 112737 000177 001114      MOV      #177,$ITEMB   ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
057764 000402      BR       1004$        ;;BRANCH OVER IBSAVE CLEARING

057766 105037 060056      1003$: CLRB      IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
057772 1004$:
057772 032777 020000 121140      BIT      #BIT13,@SWR    ;;SKIP TYPEOUT IF SET
060000 001004      BNE      20$          ;;SKIP TYPEOUTS
060002 004737 060060      JSR      PC,$ERRTYP    ;;GO TO USER ERROR ROUTINE
    
```


060006	104401	001207		TYPE	,\$CRLF	
060012			20\$:			
060012	105737	060056	2\$:	TSTB	IBSAVE	:::SEE IF IBSAVE IS LOADED
060016	001005			BNE	3\$:::BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
060020	005777	121114		TST	@SWR	:::HALT ON ERROR
060024	100002			BPL	3\$:::SKIP IF CONTINUE
060026	000000			HALT		:::HALT ON ERROR!
060030	104407			CKSWR		:::TEST FOR CHANGE IN SOFT-SWR
060032			3\$:			
060032	022737	056666	000042	CMP	#SENDAD,@#42	:::ACT-11 AUTO-ACCEPT?
060040	001001			BNE	6\$:::BRANCH IF NO
060042	000000			HALT		:::YES
060044			6\$:			
060044	105737	060056		TSTB	IBSAVE	:::SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
060050	001236			BNE	7\$:::BRANCH BACK TO CALL ORIGINAL ERROR
060052	000002			RTI		:::RETURN
060054	000000		CPSAVE:	.WORD	0	:::LOCATION TO SAVE CPU ERROR REG CONTENTS
060056	000000		IBSAVE:	.WORD	0	:::LOCATION TO SAVE ITEM BYTE

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

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

060060			\$ERRTYP:		
060060	104401	001207		TYPE	,\$SCLRF
060064	010046			MOV	R0,-(SP)
060066	005000			CLR	R0
060070	153700	001114		BISB	@#\$ITEMB,R0
060074	001004			BNE	1\$
					:::'CARRIAGE RETURN' & 'LINE FEED'
					:::SAVE R0
					:::PICKUP THE ITEM INDEX
060076	013746	001116		MOV	\$ERRPC,-(SP)
					:::IF ITEM NUMBER IS ZERO, JUST
					:::TYPE THE PC OF THE ERROR
					:::SAVE \$ERRPC FOR TYPEOUT
					:::ERROR ADDRESS
					:::GO TYPE--OCTAL ASCII(ALL DIGITS)
060102	104402			TYPOC	
060104	000456			BR	10\$
060106	122700	000177	1\$:	CMPB	#177,R0
060112	001006			BNE	1000\$
060114	113737	001102	060416	MOVB	\$STNM,PFTSTN
060122	012700	060256		MOV	#PFECH,R0
060126	000406			BR	1001\$
060130	005300		1000\$:	DEC	R0
060132	006300			ASL	R0
060134	006300			ASL	R0
060136	006300			ASL	R0
060140	062700	001304		ADD	#\$ERRTB,R0
060144	012037	060154	1001\$:	MOV	(R0)+,2\$
060150	001404			BEQ	3\$
060152	104401			TYPE	
060154	000000		2\$:	.WORD	0
060156	104401	001207		TYPE	,\$SCLRF
060162	012037	060172	3\$:	MOV	(R0)+,4\$
060166	001404			BEQ	5\$
060170	104401			TYPE	
060172	000000		4\$:	.WORD	0
060174	104401	001207		TYPE	,\$SCLRF
060200	010146		5\$:	MOV	R1,-(SP)
060202	012001			MOV	(R0)+,R1
060204	001415			BEQ	9\$
060206	012000			MOV	(R0)+,R0
060210	105720		6\$:	TSTB	(R0)+
060212	001003			BNE	7\$
060214	013146			MOV	@(R1)+,-(SP)
060216	104402			TYPOC	
060220	000402			BR	8\$
060222			7\$:	BR	8\$
060222	013146			MOV	@(R1)+,-(SP)
060224	104405			TYPDS	
060226	005711		8\$:	TST	(R1)
060230	001403			BEQ	9\$
060232	104401	060252		TYPE	,11\$
060236	000764			BR	6\$
					:::SAVE @(R1)+ FOR TYPEOUT
					:::GO TYPE--DECIMAL ASCII WITH SIGN
					:::IS THERE ANOTHER NUMBER?
					:::BR IF NO
					:::TYPE TWO(2) SPACES
					:::LOOP
060240	012601		9\$:	MOV	(SP)+,R1
060242	012600		10\$:	MOV	(SP)+,R0
					:::RESTORE R1
					:::RESTORE R0

060244	104401	001207			TYPE	,SCLRF		::"CARRIAGE RETURN" & "LINE FEED"
060250	000207				RTS	PC		::RETURN
060252	040	040	000	11\$:	.ASCIZ	/ /		::TWO(2) SPACES
					.EVEN			
060256	060266	060350	060402	PFECH:	PFECH1,PFECH2,PFECH3,PFECH4			::WORDS DEFINING TABLES BELOW
060266	120	117	127	PFECH1:	.ASCIZ	?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?		
060350	124	105	123	PFECH2:	.ASCIZ	?TESTNO ERR PC CPUERREG?		
					.EVEN			
060402	060416	001116	060054	PFECH3:	.WORD	PFTSTN,\$ERRPC,CPSAVE,0		
060412	000	000	000	PFECH4:	.BYTE	0,0,0,0		
060416	000000			PFTSTN:	.WORD	0		::CONTAINS TEST NUMBER FOR PF BIT ERROR

.SBTTL TYPE ROUTINE

```

:*****
:*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
:*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
:*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
:*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
:*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

```

```

:*CALL:
:*1) USING A TRAP INSTRUCTION
:* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:*OR
:* TYPE
:* MESADR

```

```

060420 105737 001157 $TYPE: TSTB $TFPLG ;;IS THERE A TERMINAL?
060424 100002 BPL 1$ ;;BR IF YES
060426 000000 HALT ;;HALT HERE IF NO TERMINAL
060430 000407 BR 3$ ;;LEAVE
060432 010046 1$: MOV R0,-(SP) ;;SAVE R0
060434 017600 000002 MOV @2(SP),R0 ;;GET ADDRESS OF ASCIZ STRING
060440 112046 2$: MOVB (R0)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
060442 001005 BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
060444 005726 TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
060446 012600 60$: MOV (SP)+,R0 ;;RESTORE R0
060450 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
060454 000002 RTI ;;RETURN
060456 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
060462 001430 BEQ 8$
060464 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
060470 001006 BNE 5$
060472 005726 TST (SP)+ ;;POP <CR><LF> EQUIV
060474 104401 TYPE ;;TYPE A CR AND LF
060476 001207 $CRLF
060500 105037 060706 CLR B $CHARCNT ;;CLEAR CHARACTER COUNT
060504 000755 BR 2$ ;;GET NEXT CHARACTER
060506 004737 060570 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
060512 123726 001156 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
060516 001350 BNE 2$ ;;IF NO GO GET NEXT CHAR.
060520 013746 001154 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
;;AND THE NULL CHAR.
060524 105366 000001 7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
060530 002770 BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
060532 004737 060570 JSR PC,$TYPEC ;;GO TYPE A NULL
060536 105337 060706 DECB $CHARCNT ;;DO NOT COUNT AS A COUNT
060542 000770 BR 7$ ;;LOOP

```

:HORIZONTAL TAB PROCESSOR

```

060544 112716 000040 8$: MOV B #' ,(SP) ;;REPLACE TAB WITH SPACE
060550 004737 060570 9$: JSR PC,$TYPEC ;;TYPE A SPACE
060554 132737 000007 060706 BIT B #7,$CHARCNT ;;BRANCH IF NOT AT
060562 001372 BNE 9$ ;;TAB STOP
060564 005726 TST (SP)+ ;;POP SPACE OFF STACK
060566 000724 BR 2$ ;;GET NEXT CHARACTER

```

060570				\$TYPEC:	TSTB	@\$TKS	::CHAR IN KYBD BUFFER?
060570	105777	120350			BPL	10\$::BR IF NOT
060574	100022				MOV	@\$TKB, -(SP)	::GET CHAR
060576	017746	120344			BIC	#177600, (SP)	::STRIP EXTRANEIOUS BITS
060602	042716	177600			CMPB	#\$XOFF, (SP)	::WAS CHAR XOFF
060606	122716	000023			BNE	102\$::BR IF NOT
060612	001012						
060614				101\$:	TSTB	@\$TKS	::WAIT FOR CHAR
060614	105777	120324			BPL	101\$	
060620	100375				MOVB	@\$TKB, (SP)	::GET CHAR
060622	117716	120320			BIC	#177600, (SP)	::STRIP IT
060626	042716	177600			CMPB	#\$XON, (SP)	::WAS IT XON?
060632	122716	000021			BNE	101\$::BR IF NOT
060636	001366						
060640				102\$:	TST	(SP)+	::FIX STACK
060640	005726						
060642				10\$:	TSTB	@\$TPS	::WAIT UNTIL PRINTER IS READY
060642	105777	120302			BPL	10\$	
060646	100375				MOVB	2(SP), @\$TPB	::LOAD CHAR TO BE TYPED INTO DATA REG.
060650	116677	000002	120274		CMPB	#CR, 2(SP)	::IS CHARACTER A CARRIAGE RETURN?
060656	122766	000015	000002		BNE	1\$::BRANCH IF NO
060664	001003				CLRB	\$CHARCNT	::YES--CLEAR CHARACTER COUNT
060666	105037	060706			BR	\$TYPEX	::EXIT
060672	000406				CMPB	#LF, 2(SP)	::IS CHARACTER A LINE FEED?
060674	122766	000012	000002	1\$:	BEQ	\$TYPEX	::BRANCH IF YES
060702	001402				INCB	(PC)+	::COUNT THE CHARACTER
060704	105227						::CHARACTER COUNT STORAGE
060706	000000				\$CHARCNT: .WORD	0	
060710	000207				\$TYPEX: RTS	PC	

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOS   ;;CALL FOR TYPEOUT
*   .BYTE  N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*   .BYTE  M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPON   ;;CALL FOR TYPEOUT
*
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOC   ;;CALL FOR TYPEOUT
    
```

060712	017646	000000		\$TYPOS:	MOV	2(SP),-(SP)	;;PICKUP THE MODE
060716	116637	000001	061135		MOVB	1(SP),%OFILL	;;LOAD ZERO FILL SWITCH
060724	112637	061137			MOVB	(SP)+,%OMODE+1	;;NUMBER OF DIGITS TO TYPE
060730	062716	000002			ADD	#2,(SP)	;;ADJUST RETURN ADDRESS
060734	000406				BR	\$TYPON	
060736	112737	000001	061135	\$TYPOC:	MOVB	#1,%OFILL	;;SET THE ZERO FILL SWITCH
060744	112737	000006	061137		MOVB	#6,%OMODE+1	;;SET FOR SIX(6) DIGITS
060752	112737	000005	061134	\$TYPON:	MOVB	#5,%OCNT	;;SET THE ITERATION COUNT
060760	010346				MOV	R3,-(SP)	;;SAVE R3
060762	010446				MOV	R4,-(SP)	;;SAVE R4
060764	010546				MOV	R5,-(SP)	;;SAVE R5
060766	113704	061137			MOVB	%OMODE+1,R4	;;GET THE NUMBER OF DIGITS TO TYPE
060772	005404				NEG	R4	
060774	062704	000006			ADD	#6,R4	;;SUBTRACT IT FOR MAX. ALLOWED
061000	110437	061136			MOVB	R4,%OMODE	;;SAVE IT FOR USE
061004	113704	061135			MOVB	%OFILL,R4	;;GET THE ZERO FILL SWITCH
061010	016605	000012			MOV	12(SP),R5	;;PICKUP THE INPUT NUMBER
061014	005003				CLR	R3	;;CLEAR THE OUTPUT WORD
061016	006105			1\$:	ROL	R5	;;ROTATE MSB INTO 'C'
061020	000404				BR	3\$;;GO DO MSB
061022	006105			2\$:	ROL	R5	;;FORM THIS DIGIT
061024	006105				ROL	R5	
061026	006105				ROL	R5	
061030	010503				MOV	R5,R3	
061032	006103			3\$:	ROL	R3	;;GET LSB OF THIS DIGIT
061034	105337	061136			DECB	%OMODE	;;TYPE THIS DIGIT?
061040	100016				BPL	7\$;;BR IF NO
061042	042703	177770			BIC	#177770,R3	;;GET RID OF JUNK
061046	001002				BNE	4\$;;TEST FOR 0
061050	005704				TST	R4	;;SUPPRESS THIS 0?
061052	001403				BEQ	5\$;;BR IF YES
061054	005204			4\$:	INC	R4	;;DON'T SUPPRESS ANYMORE 0'S

061056	052703	000060		BIS	#'0,R3	::MAKE THIS DIGIT ASCII
061062	052703	000040	5\$:	BIS	#',R3	::MAKE ASCII IF NOT ALREADY
061066	110337	061132		MOVB	R3,8\$::SAVE FOR TYPING
061072	104401	061132		TYPE	8\$::GO TYPE THIS DIGIT
061076	105337	061134	7\$:	DECB	\$OCNT	::COUNT BY 1
061102	003347			BGT	2\$::BR IF MORE TO DO
061104	002402			BLT	6\$::BR IF DONE
061106	005204			INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
061110	000744			BR	2\$::GO DO THE LAST DIGIT
061112	012605		6\$:	MOV	(SP)+,R5	::RESTORE R5
061114	012604			MOV	(SP)+,R4	::RESTORE R4
061116	012603			MOV	(SP)+,R3	::RESTORE R3
061120	016666	000002 000004		MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
061126	012616			MOV	(SP)+,(SP)	
061130	000002			RTI		::RETURN
061132	000		8\$:	.BYTE	0	::STORAGE FOR ASCII DIGIT
061133	000			.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
061134	000		\$OCNT:	.BYTE	0	::OCTAL DIGIT COUNTER
061135	000		\$OFILL:	.BYTE	0	::ZERO FILL SWITCH
061136	000000		\$OMODE:	.WORD	0	::NUMBER OF DIGITS TO TYPE

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

* MOV NUM,-(SP) ::PUT THE BINARY NUMBER ON THE STACK
 * TYPDS ::GO TO THE ROUTINE

061140				\$TYPDS:	MOV	R0,-(SP)	::PUSH R0 ON STACK
061140	010044				MOV	R1,-(SP)	::PUSH R1 ON STACK
061142	010146				MOV	R2,-(SP)	::PUSH R2 ON STACK
061144	010246				MOV	R3,-(SP)	::PUSH R3 ON STACK
061146	010346				MOV	R5,-(SP)	::PUSH R5 ON STACK
061150	010546				MOV	#20200,-(SP)	::SET BLANK SWITCH AND SIGN
061152	012746	020200			MOV	20(SP),R5	::GET THE INPUT NUMBER
061156	016605	000020			BPL	1\$::BR IF INPUT IS POS.
061162	100004				NEG	R5	::MAKE THE BINARY NUMBER POS.
061164	005405				MOVB	#'-,1(SP)	::MAKE THE ASCII NUMBER NEG.
061166	112766	000055	000001	1\$:	CLR	R0	::ZERO THE CONSTANTS INDEX
061174	005000				MOV	#SDBLK,R3	::SETUP THE OUTPUT POINTER
061176	012703	061354			MOVB	#',(R3)+	::SET THE FIRST CHARACTER TO A BLANK
061202	112723	000040		2\$:	CLR	R2	::CLEAR THE BCD NUMBER
061206	005002				MOV	\$DTBL(R0),R1	::GET THE CONSTANT
061210	016001	061344		3\$:	SUB	R1,R5	::FORM THIS BCD DIGIT
061214	160105				BLT	4\$::BR IF DONE
061216	002402				INC	R2	::INCREASE THE BCD DIGIT BY 1
061220	005202				BR	3\$	
061222	000774				ADD	R1,R5	::ADD BACK THE CONSTANT
061224	060105			4\$:	TST	R2	::CHECK IF BCD DIGIT=0
061226	005702				BNE	5\$::FALL THROUGH IF 0
061230	001002				TSTB	(SP)	::STILL DOING LEADING 0'S?
061232	105716				BMI	7\$::BR IF YES
061234	100407				ASLB	(SP)	::MSD?
061236	106316			5\$:	BCC	6\$::BR IF NO
061240	103003				MOVB	1(SP),-1(R3)	::YES--SET THE SIGN
061242	116663	000001	177777	6\$:	BIS	#'0,R2	::MAKE THE BCD DIGIT ASCII
061250	052702	000060		7\$:	BIS	#',R2	::MAKE IT A SPACE IF NOT ALREADY A DIGIT
061254	052702	000040			MOVB	R2,(R3)+	::PUT THIS CHARACTER IN THE OUTPUT BUFFER
061260	110223				TST	(R0)+	::JUST INCREMENTING
061262	005720				CMPL	R0,#10	::CHECK THE TABLE INDEX
061264	020027	000010			BLT	2\$::GO DO THE NEXT DIGIT
061270	002746				BGT	8\$::GO TO EXIT
061272	003002				MOV	R5,R2	::GET THE LSD
061274	010502				BR	6\$::GO CHANGE TO ASCII
061276	000764				TSTB	(SP)+	::WAS THE LSD THE FIRST NON-ZERO?
061300	105726			8\$:	BPL	9\$::BR IF NO
061302	100003				MOVB	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
061304	116663	177777	177776	9\$:	CLRB	(R3)	::SET THE TERMINATOR
061312	105013				MOV	(SP)+,R5	::POP STACK INTO R5
061314	012605				MOV	(SP)+,R3	::POP STACK INTO R3
061316	012603				MOV	(SP)+,R2	::POP STACK INTO R2
061320	012602				MOV	(SP)+,R1	::POP STACK INTO R1
061322	012601						

061324	012600			MOV	(SP)+,R0	::POP STACK INTO R0
061326	104401	061354		TYPE	\$DBLK	::NOW TYPE THE NUMBER
061332	016666	000002	000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
061340	012616			MOV	(SP)+,(SP)	
061342	000002			RTI		::RETURN TO USER
061344	023420			\$DTBL:	10000.	
061346	001750				1000.	
061350	000144				100.	
061352	000012				10.	
061354				\$DBLK:	.BLKW 4	

.SBTTL TTY INPUT ROUTINE

061364 000000
 061366 000000
 061370 009000
 061372 061373

```

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
  
```

061374 005037 061364
 061400 012737 061372 061366
 061406 013737 061366 061370
 061414 012737 061444 000060
 061422 012737 000200 000062
 061430 005777 117512
 061434 012777 000100 117502
 061442 000207

```

$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,@$TKVEC+2 ;;'BR' LEVEL 4
        TST      @$TKKB      ;;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.
  
```

061444 117746 117476
 061450 042716 177600
 061454 021627 000021
 061460 001002
 061462 005726
 061464 000002
 061466
 061466 021627 000007
 061472 001004
 061474 022737 000176 001140
 061502 001500

```

$TKSRV: MOV      @$TKB,-(SP)  ;;PICKUP THE CHARACTER
        BIC      #^C177,(SP)  ;;STRIP THE JUNK
        CMP      (SP),#$XON   ;;IS IT A RANDOM XON?
        BNE      30$          ;;BRANCH IF NO
        TST      (SP)+        ;;CLEAN RANDOM XON OFF STACK
        RTI                    ;;RETURN
30$:
1$:      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     ,SBELL       ;;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      @$TKS        ;;DISABLE TTY KEYBOARD INTERRUPTS
        TST      (SP)+        ;;CLEAN CHAR OFF STACK
31$:     TST      @$TKS        ;;WAIT FOR A CHAR
        BPL      31$         ;;LOOP UNTIL ITS THERE
  
```

061504
 061504 022737 000001 061364
 061512 001004
 061514 104401 001202
 061520 005726
 061522 000451
 061524 021627 000023
 061530 001021
 061532 005077 117406
 061536 005726
 061540 105777 117400
 061544 100375

```

061546 117746 117374      MOVB   @STKB,-(SP)      ;;GET THE CHARACTER
061552 042716 177600      BIC    #^C177,(SP)    ;;MAKE IT 7-BIT ASCII
061556 022627 000021      CMP    (SP)+,#21      ;;IS IT A CONTROL-Q?
061562 001366              BNE    31$            ;;BRANCH IF NO
061564 012777 000100 117352  MOV    #100,@STKS     ;;REENABLE TTY KEYBOARD INTERRUPTS
061572 000002              RTI                    ;;RETURN
061574 005237 061364      32$:  INC    $TKCNT       ;;COUNT THIS CHARACTER
061600 021627 000140      CMP    (SP),#140     ;;IS IT UPPER CASE?
061604 002405              BLT    4$            ;;BRANCH IF YES
061606 021627 000175      CMP    (SP),#175     ;;IS IT A SPECIAL CHAR?
061612 003002              BGT    4$            ;;BRANCH IF YES
061614 042716 000040      BIC    #40,(SP)      ;;MAKE IT UPPER CASE
061620 112677 177542      4$:  MOVB   (SP)+,@STKQIN ;;AND PUT IT IN QUEUE
061624 005237 061366      INC    $TKQIN        ;;UPDATE THE POINTER
061630 023727 061366 061373  CMP    $TKQIN,$STKQEND ;;GO OFF THE END?
061636 001003              BNE    5$            ;;BRANCH IF NO
061640 012737 061372 061366  MOV    #$STKQSRT,$TKQIN ;;RESET THE POINTER
061646 000002      5$:  RTI                    ;;RETURN

```

*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
*CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

061650 022737 000176 001140 $CKSWR: CMP    #SWREG,SWR    ;;IS THE SOFT-SWR SELECTED
061656 001104              BNE    15$           ;;EXIT IF NOT
061660 105777 117260      TSTB   @STKS         ;;IS A CHAR WAITING?
061664 100101              BPL    15$           ;;IF NOT, EXIT
061666 117746 117254      MOVB   @STKB,-(SP)   ;;YES
061672 042716 177600      BIC    #^C177,(SP)  ;;MAKE IT 7-BIT ASCII
061676 021627 000007      CMP    (SP),#7      ;;IS IT A CONTROL-G?
061702 001300              BNE    2$            ;;IF NOT, PUT IT IN THE TTY QUEUE
                                ;;AND EXIT

```

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

```

061704 123727 001134 000001 6$:  CMPB   $AUTOB,#1     ;;ARE WE RUNNING IN AUTO-MODE?
061712 001674              BEQ    2$            ;;BRANCH IF YES
061714 005726              TST    (SP)+         ;;CLEAR CONTROL-G OFF STACK
061716 004737 061374      JSR    PC,$TKINT     ;;FLUSH THE TTY INPUT QUEUE
061722 005077 117216      CLR    @STKS        ;;DISABLE TTY KEYBOARD INTERRUPTS
061726 112737 000001 001135  MOVB   #1,$INTAG     ;;SET INTERRUPT MODE INDICATOR

061734 104401 062512      TYPE   ,SCNTLG       ;;ECHO THE CONTROL-G (^G)
061740 104401 062517      TYPE   ,SMSWR        ;;TYPE CURRENT CONTENTS
061744 013746 000176      MOV    SWREG,-(SP)   ;;SAVE SWREG FOR TYPEOUT
061750 104402              TYPOC                ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
061752 104401 062530      TYPE   ,SMNEW        ;;PROMPT FOR NEW SWR
061756 005046      19$:  CLR    -(SP)         ;;CLEAR COUNTER
061760 005046              CLR    -(SP)         ;;THE NEW SWR
061762 105777 117156      7$:  TSTB   @STKS        ;;CHAR THERE?
061766 100375              BPL    7$            ;;IF NOT TRY AGAIN

061770 117746 117152      MOVB   @STKB,-(SP)   ;;PICK UP CHAR
061774 042716 177600      BIC    #^C177,(SP)  ;;MAKE IT 7-BIT ASCII

```


TTY INPUT ROUTINE

```

062204 005337 061364          DEC      $TKCNT      ;;DECREMENT THE COUNTER
062210 117766 177154 000004    MOV     @$TKQOUT,4(SP) ;;GET ONE CHARACTER
062216 005237 061370          INC     $TKQOUT      ;;UPDATE THE POINTER
062222 023727 061370 061373    CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
062230 001003                BNE     2$          ;;BRANCH IF NO
062232 012737 061372 061370    MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
062240 000002                RTI      ;;RETURN
2$:
;*****
;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
;CALL:
;*   RDLIN      ;;INPUT A STRING FROM THE TTY
;*   RETURN HERE ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
;*                ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

062242 010346          $RDLIN: MOV     R3,-(SP)      ;;SAVE R3
062244 005046          CLR     -(SP)        ;;CLEAR THE RUBOUT KEY
062246 012703 062476    1$:   MOV     #$TTYIN,R3    ;;GET ADDRESS
062252 022703 062505    2$:   CMP     #$TTYIN+7,R3  ;;BUFFER FULL?
062256 101456          BLOS   4$            ;;BR IF YES
062260 104410          RDCHR          ;;GO READ ONE CHARACTER FROM THE TTY
062262 112613          MOV     (SP)+,(R3)    ;;GET CHARACTER
062264 122713 000177    10$:  CMP     #177,(R3)     ;;IS IT A RUBOUT
062270 001022          BNE     5$            ;;BR IF NO
062272 005716          TST     (SP)          ;;IS THIS THE FIRST RUBOUT?
062274 001007          BNE     6$            ;;BR IF NO
062276 112737 000134 062474    MOV     #'\",9$      ;;TYPE A BACK SLASH
062304 104401 062474          TYPE   ,9$
062310 012716 177777          MOV     #-1,(SP)     ;;SET THE RUBOUT KEY
062314 005303          6$:   DEC     R3          ;;BACKUP BY ONE
062316 020327 062476    CMP     R3,$$TTYIN   ;;STACK EMPTY?
062322 103434          BLO    4$            ;;BR IF YES
062324 111337 062474    MOV     (R3),9$     ;;SETUP TO TYPEOUT THE DELETED CHAR.
062330 104401 062474          TYPE   ,9$
062334 000746          BR     2$            ;;GO TYPE
062336 005716          5$:   TST     (SP)          ;;RUBOUT KEY SET?
062340 001406          BEQ    7$            ;;BR IF NO
062342 112737 000134 062474    MOV     #'\",9$      ;;TYPE A BACK SLASH
062350 104401 062474          TYPE   ,9$
062354 005016          CLR     (SP)        ;;CLEAR THE RUBOUT KEY
062356 122713 000925    7$:   CMP     #25,(R3)    ;;IS CHARACTER A CTRL U?
062362 001003          BNE     8$            ;;BR IF NO
062364 104401 062505    TYPE   ,SNTLU      ;;TYPE A CONTROL 'U'
062370 000726          BR     1$            ;;GO START OVER
062372 122713 000022    8$:   CMP     #22,(R3)    ;;IS CHARACTER A '^R'?
062376 001011          BNE     3$            ;;BRANCH IF NO
062400 105013          CLRB   (R3)        ;;CLEAR THE CHARACTER
062402 104401 001207    TYPE   ,SRLF        ;;TYPE A 'CR' & 'LF'
062406 104401 062476    TYPE   ,TTYIN       ;;TYPE THE INPUT STRING
062412 000717          BR     2$            ;;GO PICKUP ANOTHER CHACTER
062414 104401 001206    4$:   TYPE   ,QUES       ;;TYPE A '?'
062420 000712          BR     1$            ;;CLEAR THE BUFFER AND LOOP
062422 111337 062474    3$:   MOV     (R3),9$     ;;ECHO THE CHARACTER
062426 104401 062474          TYPE   ,9$
062432 122723 000015    CMP     #15,(R3)+   ;;CHECK FOR RETURN
062436 001305          BNE     2$            ;;LOOP IF NOT RETURN
062440 105063 177777    CLRB   -1(R3)      ;;CLEAR RETURN (THE 15)
062444 104401 001210          TYPE   ,SLF        ;;TYPE A LINE FEED

```

062450	005726			TST	(SP)+	:::CLEAN RUBOUT KEY FROM THE STACK
062452	012603			MOV	(SP)+,R3	:::RESTORE R3
062454	011646			MOV	(SP)-,(SP)	:::ADJUST THE STACK AND PUT ADDRESS OF THE
062456	016666	000004	000002	MOV	4(SP),2(SP)	:::FIRST ASCII CHARACTER ON IT
062464	012766	062476	000004	MOV	#\$TTYIN,4(SP)	
062472	000002			RTI		:::RETURN
062474	000			9\$: .BYTE	0	:::STORAGE FOR ASCII CHAR. TO TYPE
062475	000			.BYTE	0	:::TERMINATOR
062476				.BLKB	7	:::RESERVE 7 BYTES FOR TTY INPUT
062505	136	125	015	\$CNTLU: .ASCIZ	/^U/<15><12>	:::CONTROL 'U'
062512	136	107	015	\$CNTLG: .ASCIZ	/^G/<15><12>	:::CONTROL 'G'
062517	015	012	123	\$MSWR: .ASCIZ	<15><12>/SWR = /	
062530	040	040	116	\$MNEW: .ASCIZ	/ NEW = /	
				.EVEN		

.SBTTL READ AN OCTAL NUMBER FROM THE TTY

```

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

062542	011646			\$RDOCT: MOV	(SP),-(SP)	::PROVIDE SPACE FOR THE
062544	016666	000004	000002	MOV	4(SP),2(SP)	::INPUT NUMBER
062552	010046			MOV	R0,-(SP)	::PUSH R0 ON STACK
062554	010146			MOV	R1,-(SP)	::PUSH R1 ON STACK
062556	010246			MOV	R2,-(SP)	::PUSH R2 ON STACK
062560	104411			1\$: RDLIN		::READ AN ASCII LINE
062562	012600			MOV	(SP)+,R0	::GET ADDRESS OF 1ST CHARACTER
062564	010037	062670		MOV	R0,5\$::AND SAVE IT
062570	005001			CLR	R1	::CLEAR DATA WORD
062572	005002			CLR	R2	
062574	112046			2\$: MOVB	(R0)+,-(SP)	::PICKUP THIS CHARACTER
062576	001420			BEQ	3\$::IF ZERO GET OUT
062600	122716	000060		CMPB	#'0,(SP)	::MAKE SURE THIS CHARACTER
062604	003026			BGT	4\$::IS AN OCTAL DIGIT
062606	122716	000067		CMPB	#'7,(SP)	
062612	002423			BLT	4\$	
062614	006301			ASL	R1	::*2
062616	006102			ROL	R2	
062620	006301			ASL	R1	::*4
062622	006102			ROL	R2	
062624	006301			ASL	R1	::*8
062626	006102			ROL	R2	
062630	042716	177770		BIC	#^C7,(SP)	::STRIP THE ASCII JUNK
062634	062601			ADD	(SP)+,R1	::ADD IN THIS DIGIT
062636	000756			BR	2\$::LOOP
062640	005726			3\$: TST	(SP)+	::CLEAN TERMINATOR FROM STACK
062642	010166	000012		MOV	R1,12(SP)	::SAVE THE RESULT
062646	010237	062700		MOV	R2,\$HIOCT	
062652	012602			MOV	(SP)+,R2	::POP STACK INTO R2
062654	012601			MOV	(SP)+,R1	::POP STACK INTO R1
062656	012600			MOV	(SP)+,R0	::POP STACK INTO R0
062660	000002			RTI		::RETURN
062662	005726			4\$: TST	(SP)+	::CLEAN PARTIAL FROM STACK
062664	105010			CLRB	(R0)	::SET A TERMINATOR
062666	104401			TYPE		::TYPE UP THRU THE BAD CHAR.
062670	000000			5\$: .WORD	0	
062672	104401	001206		TYPE	,\$QUES	::'?' 'CR' & 'LF'
062676	000730			BR	1\$::TRY AGAIN
062700	000000			\$HIOCT: .WORD	0	::HIGH ORDER BITS GO HERE

.SBTTL SAVE AND RESTORE R0-R5 ROUTINES

```

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

```

062702          $SAVREG:
062702 010046   MOV     R0,-(SP)      ;;PUSH R0 ON STACK
062704 010146   MOV     R1,-(SP)      ;;PUSH R1 ON STACK
062706 010246   MOV     R2,-(SP)      ;;PUSH R2 ON STACK
062710 010346   MOV     R3,-(SP)      ;;PUSH R3 ON STACK
062712 010446   MOV     R4,-(SP)      ;;PUSH R4 ON STACK
062714 010546   MOV     R5,-(SP)      ;;PUSH R5 ON STACK
062716 016646 000022   MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
062722 016646 000022   MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
062726 016646 000022   MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
062732 016646 000022   MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
062736 000002   RTI
    
```

```

*RESTORE R0-R5
*CALL:
*   RESREG
    
```

```

062740          $RESREG:
062740 012666 000022   MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
062744 012666 000022   MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
062750 012666 000022   MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
062754 012666 000022   MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
062760 012605   MOV     (SP)+,R5        ;;POP STACK INTO R5
062762 012604   MOV     (SP)+,R4        ;;POP STACK INTO R4
062764 012603   MOV     (SP)+,R3        ;;POP STACK INTO R3
062766 012602   MOV     (SP)+,R2        ;;POP STACK INTO R2
062770 012601   MOV     (SP)+,R1        ;;POP STACK INTO R1
062772 012600   MOV     (SP)+,R0        ;;POP STACK INTO R0
062774 000002   RTI
    
```


.SBTTL TRAP DECODER

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

062776 010046
 063000 016600 000002
 063004 005740
 063006 111000
 063010 006300
 063012 016000 063032
 063016 000200

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

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

063020 011646
 063022 016666 000004 000002
 063030 000002

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

.SBTTL TRAP TABLE

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

	ROUTINE	

063032 063020	\$TRPAD: .WORD \$TRAP2	
063034 060420	\$TYPE ;:CALL=TYPE	TRAP+1(104401) TTY TYPEOUT ROUTINE
063036 060736	\$TYPOC ;:CALL=TYPOC	TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
063040 060712	\$TYPOS ;:CALL=TYPOS	TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZERO)
063042 060752	\$TYPON ;:CALL=TYPON	TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
063044 061140	\$TYPDS ;:CALL=TYPDS	TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
063046 061740	\$GTSWR ;:CALL=GTSWR	TRAP+6(104406) GET SOFT-SWR SETTING
063050 061650	\$CKSWR ;:CALL=CKSWR	TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
063052 062152	\$RDCHR ;:CALL=RDCHR	TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
063054 062242	\$RDLIN ;:CALL=RDLIN	TRAP+11(104411) TTY TYPEIN STRING ROUTINE
063056 062542	\$RDOCT ;:CALL=RDOCT	TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
063060 062702	\$SAVREG ;:CALL=SAVREG	TRAP+13(104413) SAVE R0-R5 ROUTINE
063062 062740	\$RESREG ;:CALL=RESREG	TRAP+14(104414) RESTORE R0-R5 ROUTINE

.SBTTL TELETYPE MESSAGES

2				
3	063064	200	105	116 ENTERA: .ASCIZ <CRLF>/ENTER DRIVE ADDRESS: /
4	063113	040	077	111 ADRERR: .ASCIZ / ?INVALID ADDRESS/<CRLF>
5	063136	040	077	111 BADENT: .ASCIZ / ?INVALID ENTRY/<CRLF>
6	063157	200	120	117 PORTAIS: .ASCIZ <CRLF>/PORT 'A' ADDRESS IS: /
7	063206	200	120	117 PORTBIS: .ASCIZ <CRLF>/PORT 'B' ADDRESS IS: /
8	063235	200	123	131 NOCLOCK: .ASCIZ <CRLF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CRLF><LF>
9	063302	012	105	116 TESTNO: .ASCIZ <LF>/ENTER TEST #: /
10	063322	040	077	111 BADNO: .ASCIZ / ?INVALID TEST NUMBER/<CRLF>
11	063351	200	012	122 ADDRIS: .ASCIZ <CRLF><LF>@RP/RH ADDRESS (RPCS1) IS: @
12	063406	012	105	116 NTRH1: .ASCIZ <LF>@ENTER RP/RH ADDRESS: @
13	063435	040	105	122 TSTERR: .ASCIZ / ERRORS/<CRLF>

.SBTTL TEST ERROR MESSAGES

14				
15				
16				
17	063446	127	122	117 EM1: .ASCIZ /WRONG DRIVE TYPE/
18	063467	104	122	111 EM2: .ASCIZ /DRIVE NOT ON LINE/
19	063511	123	105	122 EM3: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
20	063573	104	122	111 EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/
21	063624	127	122	117 EM5: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
22	063672	122	105	107 EM6: .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
23	063772	122	105	107 EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
24	064053	122	105	107 EM10: .ASCIZ /REGISTER CONTENTS WRONG/
25	064103	103	117	116 EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
26	064167	104	122	111 EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
27	064237	122	105	101 EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
28	064324	126	117	114 EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
29	064367	101	124	124 EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
30	064446	101	124	124 EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
31	064521	101	124	124 EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
32	064600	104	122	111 EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
33	064660	104	122	111 EM21: .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
34	064733	104	122	111 EM22: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
35	065020	124	111	115 EM23: .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
36	065066	122	105	114 EM24: .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
37	065145	124	111	115 EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
38	065210	104	122	111 EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
39	065275	122	105	107 EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
40	065353	104	122	111 EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
41	065450	104	122	111 EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
42	065525	101	124	124 EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
43	065576	104	122	111 EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
44	065700	104	122	111 EM34: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
45	066003	124	111	115 EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
46	066062	124	111	115 EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
47	066134	104	122	111 EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
48	066202	101	124	124 EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
49	066257	124	111	115 EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
50	066321	104	122	111 EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
51	066407	104	122	111 EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
52	066464	127	122	111 EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
53	066541	103	117	116 EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
54	066620	103	101	116 EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
55	066667	101	124	124 EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
56	066755	101	124	124 EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
57	067040	101	124	124 EM51: .ASCIZ /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/

SYMBOL TABLE

ABS	=	000200	CKSWR	=	104407	DT06	=	000100	EM46	=	066620	MRD	=	000020
ACL	=	000040	CLOCK	=	057040	DT07	=	000200	EM47	=	066667	MSE	=	000020
ACU	=	100000	CLR	=	000040	DT08	=	000400	EM5	=	063624	MSTCK	=	000010
ADDRIS	=	063351	CPSAVE	=	060054	DT1	=	071232	EM50	=	066755	MWR	=	000040
ADRERR	=	063113	CR	=	000015	DT13	=	071334	EM51	=	067040	MXF	=	001000
AOE	=	001000	CRLF	=	000200	DT22	=	071352	EM52	=	067123	NBA	=	100000
ASR1	=	001232	CSF	=	000002	DT23	=	071364	EM53	=	067221	NED	=	010000
ATA	=	100000	CSU	=	000010	DT3	=	071246	EM54	=	067302	NEM	=	004000
ATABIT	=	071656	DCK	=	100000	DT31	=	071400	EM55	=	067375	NHS	=	002000
ATO	=	000001	DCL	=	000100	DT36	=	071412	EM56	=	067442	NOATA	=	000001
AT1	=	000002	DCU	=	000001	DT37	=	071422	EM6	=	063672	NOCLOC	=	063235
AT2	=	000004	DDISP	=	177570	DT42	=	071432	EM7	=	063772	NOSEIZ	=	001246
AT3	=	000010	DE1	=	000040	DT46	=	071440	ENTERA	=	063064	NTRH11	=	063406
AT4	=	000020	DF1	=	000002	DT5	=	071262	ERR	=	040000	OCYL	=	100000
AT5	=	000040	DF31	=	071502	DT54	=	071452	ERROR	=	104000	OFREV	=	000200
AT6	=	000100	DF36	=	071524	DT55	=	071464	ERRVEC	=	000004	OF100	=	000004
AT7	=	000200	DF42	=	071530	DT56	=	071476	EXEC	=	003010	OF200	=	000010
A16	=	000400	DF5	=	071533	DT6	=	071300	EXT1	=	000001	OF25	=	000001
A17	=	001000	DF55	=	071507	DT7	=	071314	EXT10	=	000010	OF400	=	000020
B*DENT	=	063136	DF56	=	071535	DVA	=	004000	EXT2	=	000002	OF50	=	000002
BADNO	=	063322	DF7	=	071541	ECH	=	000100	EXT20	=	000020	OF800	=	000040
BADTMO	=	002064	DH1	=	071515	ECI	=	004000	EXT4	=	000004	OPE	=	020000
BAI	=	000010	DH11	=	067504	EMTVEC	=	000030	EXT40	=	000040	OPI	=	020000
BIT0	=	000001	DH13	=	070144	EM1	=	063446	FEN	=	000200	OPPRT	=	001240
BIT00	=	000001	DH22	=	070215	EM10	=	064053	FER	=	000020	OR	=	000200
BIT01	=	000002	DH23	=	070334	EM11	=	064103	FMT22	=	010000	PAR	=	000010
BIT02	=	000004	DH26	=	070431	EM12	=	064167	F1	=	000002	PAT	=	000020
BIT03	=	000010	DH3	=	070530	EM13	=	064237	F2	=	000004	PFECH	=	060256
BIT04	=	000020	DH31	=	067555	EM14	=	064324	F3	=	000010	PFECH1	=	060266
BIT05	=	000040	DH36	=	070606	EM15	=	064367	F4	=	000020	PFECH2	=	060350
BIT06	=	000100	DH4	=	070704	EM16	=	064446	F5	=	000040	PFECH3	=	060402
BIT07	=	000200	DH42	=	067624	EM17	=	064521	GO	=	000001	PFECH4	=	060412
BIT08	=	000400	DH44	=	070733	EM2	=	063467	GRV	=	000010	PFTSTN	=	060416
BIT09	=	001000	DH46	=	070752	EM20	=	064600	GTSWR	=	104406	PGE	=	002000
BIT1	=	000002	DH5	=	071047	EM21	=	064660	HCE	=	000200	PGM	=	001000
BIT10	=	002000	DH55	=	067746	EM22	=	064733	HCI	=	002000	PIP	=	020000
BIT11	=	004000	DH56	=	071144	EM23	=	065020	HCRC	=	000400	PIRG	=	177772
BIT12	=	010000	DH7	=	071222	EM24	=	065066	HT	=	000011	PIRQVE	=	000240
BIT13	=	020000	DIGB	=	070022	EM25	=	065145	IAE	=	002000	PLU	=	020000
BIT14	=	040000	DISPLA	=	000004	EM26	=	065210	IBSAVE	=	060056	PORTA	=	001224
BIT15	=	100000	DISPRE	=	001142	EM27	=	065275	IE	=	000100	PORTAI	=	063157
BIT2	=	000004	DLT	=	000174	EM3	=	063511	ILF	=	000001	PORTB	=	001226
BIT3	=	000010	DL64	=	100000	EM30	=	065353	ILR	=	000002	PORTBI	=	063206
BIT4	=	000020	DMD	=	000020	EM31	=	065450	IOTVEC	=	000020	PORTC	=	001230
BIT5	=	000040	DPR	=	000001	EM32	=	065525	IR	=	000100	PRE	=	000020
BIT6	=	000100	DRQ	=	000400	EM33	=	065576	IXE	=	004000	PRO	=	000000
BIT7	=	000200	DRY	=	004000	EM34	=	065700	KYBCTL	=	001274	PR1	=	000040
BIT8	=	000400	DSWR	=	000200	EM35	=	066003	LF	=	000012	PR2	=	000100
BIT9	=	001000	DTE	=	177570	EM36	=	066062	LST	=	002000	PR3	=	000140
BPTVEC	=	000014	DT00	=	010000	EM37	=	066134	MAXTN	=	071666	PR4	=	000200
CHANGE	=	003174	DT01	=	000200	EM4	=	063573	MCLK	=	000002	PR5	=	000240
CHGADR	=	001276	DT02	=	000001	EM40	=	066202	MCPE	=	020000	PR6	=	000300
CKCLK	=	056706	DT03	=	000002	EM41	=	066257	MHS	=	001000	PR7	=	000340
CKCLK1	=	056756	DT04	=	000004	EM42	=	066321	MINX	=	000004	PS	=	177776
CKCLK2	=	057020	DT05	=	000010	EM43	=	066407	MOH	=	020000	PSEL	=	002000
CKCLK3	=	057030				EM44	=	066464	MOL	=	010000	PSU	=	000001
CKERR	=	001244				EM45	=	066541	MPE	=	000400	PSW	=	177776

SYMBOL TABLE

PTNBR	=	001234	SW04	=	000020	TEST44	=	053726	TST40	=	046774	\$ENULL	=	056702
PURVEC	=	000024	SW05	=	000040	TEST45	=	054614	TST41	=	047756	\$EOP	=	056460
RAW	=	000020	SW06	=	000100	TEST46	=	055562	TST41B	=	051032	\$EOPCT	=	056516
RDCHR	=	104410	SW07	=	000200	TEST5	=	011132	TST42	=	051364	\$ERFLG	=	001103
RDLIN	=	104411	SW08	=	000400	TEST6	=	012270	TST42B	=	052440	\$ERMAX	=	001115
RDOCT	=	104412	SW09	=	001000	TEST7	=	013260	TST43	=	052772	\$ERROR	=	057532
RDY	=	000200	SW1	=	000002	TIME	=	001252	TST44	=	053654	\$ERRPC	=	001116
RELERR	=	001250	SW10	=	002000	TIMEA	=	001256	TST45	=	054542	\$ERRTB	=	001304
RELOK	=	000001	SW11	=	004000	TIMEAM	=	001262	TST46	=	055510	\$ERRTY	=	060060
RESREG	=	104414	SW12	=	010000	TIMEAP	=	001260	TST47	=	056456	\$ERTTL	=	001112
RESVEC	=	000010	SW13	=	020000	TIMEB	=	001264	TST5	=	011060	\$ESCAP	=	001200
RMR	=	000004	SW14	=	040000	TIMEBM	=	001270	TST6	=	012216	\$FILLC	=	001156
RPAS	=	000016	SW15	=	100000	TIMEBP	=	001266	TST7	=	013206	\$FILLS	=	001155
RPBA	=	000004	SW2	=	000004	TIMES	=	001272	TUF	=	000100	\$GDADR	=	001120
RPCA	=	000034	SW3	=	000010	TKVEC	=	000060	TYPDS	=	104405	\$GDDAT	=	001124
RPCC	=	000036	SW4	=	000020	TOLER	=	057072	TYPE	=	104401	\$GET42	=	056656
RPCS1	=	000000	SW5	=	000040	TPVEC	=	000064	TYPOC	=	104402	\$GTSWR	=	061740
RPCS2	=	000010	SW6	=	000100	TRAPVE	=	000034	TYPON	=	104404	\$GT42P	=	056652
RPDA	=	000006	SW7	=	000200	TRE	=	040000	TYPOS	=	104403	\$HD	=	000000
RPDB	=	000022	SW8	=	000400	TRK1	=	004000	UNS	=	040000	\$HIOCT	=	062700
RPDS1	=	000012	SW9	=	001000	TRK10	=	040000	UPE	=	020000	\$ICNT	=	001104
RPDT	=	000026	TAP	=	040000	TRK2	=	010000	US1	=	000001	\$INTAG	=	001135
RPEC1	=	000044	TBITVE	=	000014	TRK20	=	100000	US2	=	000002	\$ITEMB	=	001114
RPEC2	=	000046	TDF	=	000040	TRK4	=	020000	US4	=	000004	\$LF	=	001210
RPER1	=	000014	TESTNO	=	063302	TRTVEC	=	000014	UWR	=	000010	\$LKCSB	=	001214
RPER2	=	000040	TEST1	=	003374	TSTADR	=	071542	VUF	=	000002	\$LKCSR	=	001212
RPER3	=	000042	TEST10	=	014250	TSTERR	=	063435	VU30	=	010000	\$LKS	=	001220
RPLA	=	000020	TEST11	=	014672	TSTNUM	=	001242	VV	=	000100	\$LLVEC	=	001222
RPMR	=	000024	TEST12	=	015314	TST1	=	003322	VVSET	=	000001	\$LPADR	=	001106
RPOF	=	000032	TEST13	=	016672	TST1AA	=	003304	WAO	=	000002	\$LPERR	=	001110
RPSN	=	000030	TEST14	=	020250	TST10	=	014176	WATCH	=	001254	\$LPVEC	=	001216
RPWC	=	000002	TEST15	=	021552	TST11	=	014620	WCE	=	040009	\$MNEW	=	062530
R6	=	X000006	TEST16	=	023054	TST12	=	015242	WCF	=	000040	\$MSWR	=	062517
R7	=	X000007	TEST17	=	024314	TST13	=	016620	WCU	=	000001	\$MXCNT	=	057530
SAVREG	=	104413	TEST2	=	004774	TST14	=	020176	WLE	=	004000	\$NULL	=	001154
SC	=	100000	TEST20	=	025202	TST15	=	021500	WRL	=	004000	\$NWTST	=	000000
SCOPE	=	000004	TEST21	=	026070	TST16	=	023002	WRU	=	000400	\$OCNT	=	061134
SC1	=	000100	TEST22	=	027250	TST17	=	024242	WSU	=	000004	\$OMODE	=	061136
SC10	=	001000	TEST23	=	030430	TST2	=	004722	\$AUTOB	=	001134	\$OVER	=	057514
SC2	=	000200	TEST24	=	031440	TST20	=	025130	\$BDADR	=	001122	\$PASS	=	001100
SC20	=	002000	TEST25	=	032450	TST21	=	026016	\$BDDAT	=	001126	\$QUES	=	001206
SC4	=	000400	TEST26	=	033656	TST22	=	027176	\$BELL	=	001202	\$RDCHR	=	062152
SEIZPT	=	001236	TEST27	=	035064	TST23	=	030356	\$CHARC	=	060706	\$RDLIN	=	062242
SKI	=	040000	TEST3	=	006374	TST24	=	031366	\$CKSWR	=	061650	\$RDOCT	=	062542
STA	=	002554	TEST30	=	037060	TST25	=	032376	\$CMTAG	=	001100	\$RDSZ	=	000007
STACK	=	001100	TEST31	=	037554	TST26	=	033604	\$CM1	=	000001	\$REGAD	=	001160
START	=	002144	TEST32	=	041014	TST27	=	035012	\$CM2	=	000002	\$REGJ	=	001162
START1	=	002154	TEST33	=	042254	TST3	=	006322	\$CM3	=	000001	\$RESRE	=	062740
START2	=	002162	TEST34	=	043344	TST30	=	037006	\$CM4	=	000005	\$RPADR	=	001300
STKLMT	=	177774	TEST35	=	044434	TST31	=	037502	\$CNTLG	=	062512	\$RPVEC	=	001302
SWR	=	001140	TEST36	=	045250	TST32	=	040742	\$CNTLU	=	062505	\$RTNAD	=	056700
SWREG	=	000176	TEST37	=	046064	TST33	=	042202	\$CRLF	=	001207	\$SAVRE	=	062702
SWO	=	000001	TEST4	=	007774	TST34	=	043272	\$DBLK	=	061354	\$SCOPE	=	057132
SWO0	=	000001	TEST40	=	047046	TST35	=	044362	\$DOAGN	=	056676	\$SETUP	=	000127
SWO1	=	000002	TEST41	=	050030	TST36	=	045176	\$DTBL	=	061344	\$STUP	=	177777
SWO2	=	000004	TEST42	=	051436	TST37	=	046012	\$ENDAD	=	056666	\$SVLAD	=	057466
SWO3	=	000010	TEST43	=	053044	TST4	=	007722	\$ENDCT	=	056524	\$SVPC	=	000210

SYMBOL TABLE

\$SWR = 166000	\$TKQOU 061370	\$TMP4 001174	\$TRPAD 063032	\$TYPON 060752
\$SWRMK= 000000	\$TKQSR 061372	\$TN = 000050	\$TSTNM 001102	\$TYPOS 060712
\$TIMES 001176	\$TKS 001144	\$TPB 001152	\$TTYIN 062476	\$XOFF = 000023
\$TKB 001146	\$TKSRV 061444	\$TPFLG 001157	\$TYPDS 061140	\$XON = 000021
\$TKCNT 061364	\$TMP0 001164	\$TPS 001150	\$TYPE 060420	\$XTSTR 057150
\$TKINT 061374	\$TMP1 001166	\$TRAP 062776	\$TYPEC 060570	\$GET4= 000000
\$TKQEN= 061373	\$TMP2 001170	\$TRAP2 063020	\$TYPEX 060710	\$OFILL 061135
\$TKQIN 061366	\$TMP3 001172	\$TRP = 000015	\$TYPOC 060736	

. ABS. 071670 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 60416 WORDS (236 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
CZRJED.BIN,CZRJED/C=CZRJED.DOC,CZRJED,[20,0]SYSMAC/M

\$GET4	15-1	15-1#	21-1*	21-1*										
\$OFILL	21-1	21-1#												
\$4OCAT	17-1	18-1												
\$AUTOB	7-0#	11-29*	23-1	23-1	23-1									
\$BDADR	7-0#	13-31*	13-31*	13-31*	13-31*	13-38*	13-38*	13-38*	13-38*	13-46*	13-46*	13-46*	13-46*	13-46*
	13-46*	13-46*	13-46*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*
	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*
	13-147*	13-147*	13-147*	13-147*	13-147*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*
	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*
	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*
	13-234*	13-234*	13-234*	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*	13-233*	13-233*	13-233*	13-233*	13-234*
	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-257*	13-257*	13-257*	13-257*	13-299*
	13-352*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-314*	13-314*	13-314*	13-314*	13-352*
	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-406*	13-406*	13-406*	13-406*	13-428*
	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-428*	13-428*	13-428*	13-483*	13-483*
	13-504*	13-504*	13-504*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-504*	13-504*	13-504*	13-504*	13-504*
	13-579*	13-579*	13-579*	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-504*	13-504*	13-504*	13-504*	13-579*
	13-644*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	13-644*	13-644*	13-644*	13-644*	13-644*
	14-54*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-54*	14-54*	14-54*	14-54*	14-54*
	14-117*	14-117*	14-117*	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-117*	14-117*	14-117*	14-117*	14-117*
	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-139*	14-139*	14-139*	14-139*	14-206*
	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-206*	14-206*	14-206*	14-206*	14-302*
	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*
	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-374*	14-374*	14-374*	14-374*	14-397*
	14-518*	14-518*	14-518*	14-537*	14-537*	14-537*	14-537*	14-537*	14-537*	14-451*	14-451*	14-470*	14-470*	14-518*
	14-601*	14-601*	14-601*	14-601*	14-601*	14-601*	14-601*	14-601*	14-601*	14-537*	14-537*	14-600*	14-600*	14-601*
	14-629*	14-630*	14-630*	14-630*	14-630*	14-630*	14-630*	14-630*	14-630*	14-601*	14-601*	14-601*	14-601*	14-629*
	14-679*	14-679*	14-679*	14-695*	14-695*	14-695*	14-695*	14-695*	14-695*	14-601*	14-601*	14-601*	14-601*	14-679*
	14-789*	14-789*	14-789*	14-789*	14-789*	14-789*	14-789*	14-789*	14-789*	14-630*	14-630*	14-630*	14-630*	14-789*
\$BDDAT	7-0#	13-31	13-31	13-31	13-31	13-31	13-31	13-31	13-31	29-1	29-2	29-3	29-4	29-5
	13-46	13-46	13-46	13-46	13-46	13-46	13-46	13-46	13-46	13-38	13-38	13-38	13-38	13-38
	13-147	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-38	13-46*	13-46*	13-46*	13-46*
	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-38	13-46*	13-46*	13-46*	13-46*
	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-38	13-46*	13-46*	13-46*	13-46*
	13-233	13-233	13-233	13-233	13-233	13-233	13-233	13-233	13-233	13-168*	13-168*	13-168*	13-168*	13-168*
	13-253	13-253	13-253	13-253	13-253	13-253	13-253	13-253	13-253	13-168*	13-168*	13-168*	13-168*	13-233
	13-257*	13-299	13-299	13-299	13-299	13-299	13-299	13-299	13-299	13-168*	13-168*	13-168*	13-168*	13-257*
	13-314	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-234	13-234	13-234	13-234	13-257*
	13-352*	13-406	13-406	13-406	13-406	13-406	13-406	13-406	13-406	13-257	13-257	13-257	13-257	13-314
	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-299*	13-299*	13-299*	13-299*	13-314
	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-340*	13-340*	13-340*	13-340*	13-352*
	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-406	13-406	13-406	13-406	13-406*
	13-483*	13-504	13-504	13-504	13-504	13-504	13-504	13-504	13-504	13-428	13-428	13-428	13-428	13-428
	13-504*	13-504*	13-504*	13-555	13-555	13-555	13-555	13-555	13-555	13-428	13-428	13-428	13-428	13-483
	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-483	13-483	13-483	13-483	13-483*
	13-594	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-504	13-504	13-504	13-504	13-504*
	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-504*	13-504*	13-504*	13-504*	13-555*
	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	13-555*	13-555*	13-555*	13-555*	13-579*
	14-54*	14-54*	14-73	14-73	14-73	14-73	14-73	14-73	14-73	13-579	13-579	13-579	13-579	13-594
	14-117	14-117	14-117	14-117	14-117*	14-117*	14-117*	14-117*	14-117*	13-594*	13-594*	13-644	13-644	13-644*
	14-139	14-139	14-139	14-139	14-139	14-139	14-139	14-139	14-139	13-644	13-644	13-644	13-644	13-644*
	14-139*	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-19	14-19	14-19	14-19	14-19*
	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-54	14-54	14-54	14-54	14-54*
	14-302	14-302	14-302	14-302	14-302*	14-302*	14-302*	14-302*	14-302*	14-73	14-73	14-73	14-73	14-73*
	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-117	14-117	14-117	14-117	14-117*
	14-374	14-374	14-374	14-374*	14-374*	14-374*	14-374*	14-374*	14-374*	14-139	14-139	14-139	14-139	14-139*
										14-206	14-206	14-206	14-206	14-206*
										14-237*	14-237*	14-237*	14-237*	14-302
										14-302*	14-302*	14-302*	14-302*	14-302*
										14-319*	14-319*	14-319*	14-319*	14-319
										14-319*	14-319*	14-319*	14-319*	14-374
										14-397	14-397	14-397	14-397	14-397

	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-451*	14-451*	14-451*	14-451*	14-470*	14-470*	14-470*
	14-470*	14-518	14-518	14-518	14-518	14-518	14-518	14-518	14-518	14-518	14-518	14-537	14-537	14-537
	14-537	14-537*	14-537*	14-537*	14-537*	14-537*	14-537*	14-600	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*
	14-601	14-601	14-601	14-601	14-601	14-601	14-601	14-601*	14-601*	14-601*	14-601*	14-601*	14-601*	14-629
	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-630	14-630	14-630	14-630	14-630	14-630*	14-630*
	14-630*	14-630*	14-630*	14-679	14-679*	14-679*	14-679*	14-679*	14-679*	14-679*	14-679*	14-695	14-695*	14-695*
	14-695*	14-765	14-765	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-789	14-789*	14-789*
	14-789*	14-789*	14-789*	29-1	29-2	29-3	29-4	29-5	29-6	29-8	29-8	14-789*	14-789*	14-789*
\$BELL	7-0#	18-1	18-1	18-1	23-1	23-1	23-1	23-1						
\$CHARC	20-1	20-1#	20-1*	20-1*	20-1*									
\$CKSWR	23-1#	26-1	26-1											
\$CM1	7-0	7-0	7-0#	7-0#										
\$CM2	7-0	7-0	7-0#	7-0#										
\$CM3	7-0	7-0	7-0#											
\$CM4	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0#	7-0#	7-0#	7-0#
	7-0#	7-0#												
\$CMTAG	7-0#	11-24	11-24	11-24	11-24	11-24	11-24	11-24						
\$CNTLG	23-1	23-1#												
\$CNTLU	23-1	23-1	23-1#											
\$CRLF	7-0#	12-29	12-48	12-86	15-1	18-1	18-1	18-1	19-1	19-1	19-1	19-1	20-1	20-1
	20-1	23-1	23-1	23-1	23-1	24-1	24-1							
\$DBLK	22-1	22-1	22-1#											
\$DOAGN	15-1	15-1	15-1#											
\$DTBL	22-1	22-1#												
\$ENDAD	6-5	11-29	15-1#	18-1										
\$ENDCT	11-24	15-1#												
\$ENULL	15-1#													
\$EOP	14-696	15-1#												
\$EOPCT	11-24*	15-1	15-1#											
\$ERFLG	7-0#	17-1	17-1	17-1	17-1	17-1	17-1*	18-1	18-1	18-1*				
\$ERMAX	7-0#	11-24*	12-56*	17-1	17-1	17-1	17-1*							
\$ERROR	11-24	18-1#												
\$ERRPC	7-0#	18-1	18-1	18-1	18-1*	18-1*	19-1	19-1	29-1	29-2	29-3	29-4	29-5	29-6
	29-7	29-8	29-9	29-10	29-11	29-12	29-13	29-14	29-15					
\$ERRTB	9-0#	19-1												
\$ERRTY	18-1	19-1#												
\$ERTTL	7-0#	15-1	15-1	15-1*	18-1	18-1	18-1*							
\$ESCAP	7-0#	11-24*	17-1*											
\$FILLC	7-0#	20-1	20-1	20-1										
\$FILLS	7-0#	20-1	20-1											
\$GDADR	7-0#													
\$GDDAT	7-0#	13-31	13-31	13-31*	13-31*	13-31*	13-31*	13-38	13-38	13-38*	13-38*	13-38*	13-38*	13-46
	13-46	13-46	13-46	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-51*	13-147
	13-147	13-147	13-147	13-147	13-147*	13-147*	13-147*	13-168	13-168	13-168	13-168	13-168	13-168	13-168*
	13-168*	13-233	13-233	13-233	13-233*	13-233*	13-233*	13-233*	13-233*	13-233*	13-233*	13-233*	13-233*	13-233*
	13-234*	13-253	13-253	13-253	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*	13-253*
	13-257*	13-299	13-299	13-299	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*
	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-340	13-340	13-340*	13-340*	13-340*	13-340*	13-340*
	13-406	13-406	13-406	13-406	13-406	13-406	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*
	13-406*	13-406*	13-406*	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428*
	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*
	13-483	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*
	13-504	13-504	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*	13-504*
	13-555	13-555	13-555	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*
	13-579	13-579	13-579	13-579*	13-579*	13-579*	13-579*	13-594	13-594	13-594	13-594	13-594	13-594*	13-594*
	13-644	13-644	13-644	13-644	13-644	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	14-19	14-19

C
C
R
R
R
R
R
R
R
R

R

	14-19	14-19	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-54	14-54	14-54	14-54	14-54	14-54
	14-54*	14-54*	14-54*	14-54*	14-54*	14-73	14-73	14-73	14-73	14-73	14-73	14-73*	14-73*	14-73*
	14-73*	14-73*	14-117	14-117	14-117	14-117	14-117	14-117	14-117	14-117*	14-117*	14-117*	14-117*	14-117*
	14-139	14-139	14-139	14-139	14-139	14-139	14-139	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-206
	14-206	14-206	14-206	14-206	14-206	14-206	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*
	14-237	14-237	14-237*	14-302	14-302	14-302	14-302	14-302	14-302	14-302*	14-302*	14-302*	14-302*	14-302*
	14-302*	14-302*	14-302*	14-319	14-319	14-319	14-319	14-319	14-319	14-319*	14-319*	14-319*	14-319*	14-319*
	14-319*	14-319*	14-319*	14-374	14-374	14-374	14-374	14-374	14-374*	14-374*	14-374*	14-374*	14-374*	14-397
	14-397	14-397	14-397	14-397	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-451	14-451	14-451*	14-470
	14-470*	14-518	14-518	14-518	14-518	14-518*	14-518*	14-518*	14-518*	14-518*	14-537	14-537	14-537	14-537
	14-537*	14-537*	14-537*	14-537*	14-537*	14-600	14-600	14-600	14-600*	14-600*	14-601	14-601	14-601	14-601*
	14-601*	14-601*	14-629	14-629	14-629	14-629*	14-629*	14-629*	14-630	14-630	14-630*	14-630*	14-630*	14-679
	14-679	14-679	14-679*	14-679*	14-695	14-695	14-695	14-695*	14-695*	14-695*	14-765	14-765	14-765	14-765*
	14-765*	14-765*	14-789	14-789	14-789	14-789	14-789	14-789*	14-789*	14-789*	29-2	29-3	29-5	
\$GET42	15-1#													
\$GT42P	15-1	15-1#												
\$GT5WR	23-1#	26-1	26-1											
\$HD	5-498	5-498	5-498											
\$HIOCT	24-1#	24-1*												
\$ICNT	7-0#	12-74*	17-1	17-1	17-1	17-1*	17-1*							
\$INTAG	7-0#	23-1	23-1	23-1	23-1*									
\$ITEMB	7-0#	18-1	18-1	18-1	18-1	18-1*	18-1*	19-1						
\$LF	7-0#	18-1	18-1	20-1	20-1	23-1	23-1	23-1	24-1	24-1				
\$LKCSB	8-0#	16-12*												
\$LKCSR	8-0#	16-8	16-13*											
\$LKS	8-0#	16-17	16-21*											
\$LLVEC	8-0#	16-18												
\$LPADR	7-0#	11-24*	12-57*	13-19*	13-80*	13-166*	13-188*	13-251*	13-271*	13-312*	13-325*	13-350*	13-376*	13-426*
	13-449*	13-502*	13-524*	13-569*	13-592*	13-615*	14-17*	14-37*	14-71*	14-94*	14-137*	14-156*	14-220*	14-253*
	14-317*	14-341*	14-395*	14-415*	14-468*	14-490*	14-535*	14-564*	14-627*	14-645*	14-693*	14-723*	14-787*	17-1
\$LPERR	7-0#	11-24*	12-58*	13-19*	13-80*	13-166*	13-188*	13-251*	13-271*	13-312*	13-325*	13-350*	13-376*	13-426*
	13-449*	13-502*	13-524*	13-569*	13-592*	13-615*	14-17*	14-37*	14-71*	14-94*	14-137*	14-156*	14-220*	14-253*
	14-317*	14-341*	14-395*	14-415*	14-468*	14-490*	14-535*	14-564*	14-627*	14-645*	14-679*	14-693*	14-723*	14-723*
	14-787*	17-1	17-1	17-1	17-1*									
\$LPVEC	8-0#	16-9												
\$MAIL	11-24	11-29	17-1	18-1	20-1									
\$MNEW	23-1	23-1#												
\$MSWR	23-1	23-1#												
\$MXCNT	17-1	17-1	17-1	17-1#										
\$NULL	7-0#	20-1	20-1	20-1										
\$NWTST	13-19	13-19	13-19#	13-19#	13-80	13-80	13-80#	13-80#	13-166	13-166	13-166#	13-166#	13-188	13-188
	13-188#	13-188#	13-251	13-251	13-251#	13-251#	13-271	13-271	13-271#	13-271#	13-312	13-312	13-312#	13-312#
	13-325	13-325	13-325#	13-325#	13-350	13-350	13-350#	13-350#	13-376	13-376	13-376#	13-376#	13-426	13-426
	13-426#	13-426#	13-449	13-449	13-449#	13-449#	13-502	13-502	13-502#	13-502#	13-524	13-524	13-524#	13-524#
	13-569	13-569	13-569#	13-569#	13-592	13-592	13-592#	13-592#	13-615	13-615	13-615#	13-615#	14-17	14-17
	14-17#	14-17#	14-37	14-37	14-37#	14-37#	14-71	14-71	14-71#	14-71#	14-94	14-94	14-94#	14-94#
	14-137	14-137	14-137#	14-137#	14-156	14-156	14-156#	14-156#	14-220	14-220	14-220#	14-220#	14-253	14-253
	14-253#	14-253#	14-317	14-317	14-317#	14-317#	14-341	14-341	14-341#	14-341#	14-395	14-395	14-395#	14-395#
	14-415	14-415	14-415#	14-415#	14-468	14-468	14-468#	14-468#	14-490	14-490	14-490#	14-490#	14-535	14-535
	14-535#	14-535#	14-564	14-564	14-564#	14-564#	14-627	14-627	14-627#	14-627#	14-645	14-645	14-645#	14-645#
	14-693	14-693	14-693#	14-693#	14-723	14-723	14-723#	14-723#	14-787	14-787	14-787#	14-787#	14-794#	
\$OCNT	21-1#	21-1*	21-1*											
\$OMODE	21-1	21-1#	21-1*	21-1*	21-1*	21-1*								
\$OVER	17-1	17-1	17-1	17-1#										
\$PASS	7-0#	12-55*	15-1	15-1	15-1	15-1*	15-1*	17-1	17-1	17-1				

\$QUES	7-0#	18-1	18-1	20-1	20-1	23-1	23-1	23-1	23-1	24-1	24-1	24-1			
\$R2A	26-1														
\$RDCHR	23-1#	26-1	26-1												
\$RDDEC	26-1														
\$RDLIN	23-1#	26-1	26-1												
\$RDOCT	24-1#	26-1	26-1												
\$RDSZ	23-1	23-1#													
\$REGO	7-0#														
\$REGAD	7-0#														
\$RESRE	25-1#	26-1													
\$RPADR	8-0#	12-49	12-84	12-91*	12-94	13-16	29-16								
\$RPVEC	8-0#														
\$RTNAD	15-1#														
\$SAVRE	25-1#	26-1	26-1												
\$SCOPE	11-24	17-1#													
\$SETUP	10-192	10-192	10-192	10-192	10-192	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	11-24	11-24	11-24
	11-24	11-24	11-24	11-24	11-24	11-24	11-24	11-24	11-24	11-29	11-29	11-29	15-1	15-1	
	17-1	18-1	18-1	18-1	18-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1			
\$STUP	10-192	10-192	10-192	10-192	10-192	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#	10-192#
	10-192#														
\$SVLAD	17-1	17-1#													
\$SVPC	6-5	6-5#													
\$SWR	5-494#	5-498	5-499	5-499	5-499	5-499	5-499	5-499	5-499	5-499	7-0	7-0	7-0	11-24	
	11-24	11-24	11-24	11-24	13-19	13-80	13-166	13-188	13-251	13-271	13-312	13-325	13-350	13-376	
	13-426	13-449	13-502	13-524	13-569	13-592	13-615	14-17	14-37	14-71	14-94	14-137	14-156	14-220	
	14-253	14-317	14-341	14-395	14-415	14-468	14-490	14-535	14-564	14-627	14-645	14-693	14-723	14-787	
	14-794	15-1	15-1	15-1	15-1	15-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	
	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-1	17-3#	18-1	18-1
	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1
\$SWRMK	17-1														
\$TESTN	19-1														
\$TIMES	7-0#	11-24*	13-19*	13-80*	13-166*	13-188*	13-251*	13-271*	13-312*	13-325*	13-350*	13-376*	13-426*	13-449*	
	13-502*	13-524*	13-569*	13-592*	13-615*	14-17*	14-37*	14-71*	14-94*	14-137*	14-156*	14-220*	14-253*	14-317*	
	14-341*	14-395*	14-415*	14-468*	14-490*	14-535*	14-564*	14-627*	14-645*	14-693*	14-723*	14-787*	15-1*	17-1	
	17-1	17-1	17-1*	17-1*											
\$TKB	7-0#	20-1	20-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1	23-1	
\$TKCNT	23-1	23-1	23-1#	23-1*	23-1*	23-1*									
\$TKINT	12-6	12-53	23-1	23-1#											
\$TKQEN	23-1	23-1	23-1#	23-1*	23-1*	23-1*	23-1*								
\$TKQIN	23-1	23-1	23-1#	23-1*	23-1*	23-1*	23-1*								
\$TKQOU	23-1	23-1	23-1#	23-1*	23-1*	23-1*									
\$TKQSR	23-1	23-1	23-1	23-1#											
\$TKS	7-0#	20-1	20-1	23-1	23-1	23-1	23-1	23-1	23-1*	23-1*	23-1*	23-1*	23-1*	23-1*	
\$TKSRV	23-1	23-1#													
\$TMPO	7-0#	13-31	13-31	13-31*	13-31*	13-31*	13-31*	13-38	13-38	13-38*	13-38*	13-38*	13-38*	13-38*	13-46
	13-46	13-46	13-46	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-46*	13-147	13-147	13-147
	13-147	13-147*	13-147*	13-147*	13-147*	13-168	13-168	13-168	13-168	13-168*	13-168*	13-168*	13-168*	13-168*	13-233
	13-233	13-233*	13-233*	13-233*	13-233*	13-234	13-234	13-234	13-234	13-234*	13-234*	13-234*	13-234*	13-234*	13-253
	13-253	13-253*	13-253*	13-253*	13-253*	13-257	13-257	13-257	13-257	13-257*	13-257*	13-257*	13-257*	13-257*	13-299
	13-299	13-299	13-299	13-299	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-299*	13-314	13-314	13-314
	13-314	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-340	13-340	13-340	13-340*	13-340*	13-340*	13-340*	13-352
	13-352	13-352*	13-352*	13-406	13-406	13-406	13-406	13-406	13-406	13-406	13-406	13-406*	13-406*	13-406*	13-406*
	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-428	13-428	13-428	13-428	13-428	13-428
	13-428	13-428	13-428	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*
	13-428*	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*
	13-483*	13-483*	13-483*	13-483*	13-504	13-504	13-504	13-504	13-504	13-504	13-504	13-504	13-504*	13-504*	13-504*

DT56	10-189	29-16#					
DT6	29-4#						
DT7	9-27	9-48	9-90	9-97	9-104	10-28	29-5#
DTE	5-591#						
DTSY	5-604#						
DVA	5-556#						
ECH	5-585#						
ECI	5-699#						
EM1	9-4	27-17#					
EM10	9-53	27-24#					
EM11	9-60	27-25#					
EM12	9-67	27-26#					
EM13	9-74	27-27#					
EM14	9-81	27-28#					
EM15	9-88	27-29#					
EM16	9-95	27-30#					
EM17	9-102	27-31#					
EM2	9-11	27-18#					
EM20	9-109	27-32#					
EM21	9-116	27-33#					
EM22	9-123	27-34#					
EM23	9-130	27-35#					
EM24	10-3	27-36#					
EM25	10-11	27-37#					
EM26	10-19	27-38#					
EM27	10-26	27-39#					
EM3	9-18	27-19#					
EM30	10-33	27-40#					
EM31	10-40	27-41#					
EM32	10-47	27-42#					
EM33	10-54	27-43#					
EM34	10-61	27-44#					
EM35	10-68	27-45#					
EM36	10-75	27-46#					
EM37	10-82	27-47#					
EM4	9-25	27-20#					
EM40	10-89	27-48#					
EM41	10-96	27-49#					
EM42	10-103	27-50#					
EM43	10-110	27-51#					
EM44	10-117	27-52#					
EM45	10-124	27-53#					
EM46	10-131	27-54#					
EM47	10-138	27-55#					
EM5	9-32	27-21#					
EM50	10-145	27-56#					
EM51	10-152	27-57#					
EM52	10-159	27-58#					
EM53	10-166	27-59#					
EM54	10-173	27-60#					
EM55	10-180	27-61#					
EM56	10-187	27-62#					
EM6	9-39	27-22#					
EM7	9-46	27-23#					
EMTVEC	5-501#	11-24*					
ENTERA	12-11	27-3#					

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
0
1
2
3
4
5
6
7
8
9
*
#

ERR	5-574#	14-302	14-319											
ERROR	5-501#													
ERRVEC	5-501#	11-24	11-24*	11-24*	11-26*	11-27*	16-6*	16-7*	16-16*	16-25*	17-1	17-1	17-1*	17-1*
	17-1*	17-1*	17-1*	18-1	18-1*	18-1*								
EXEC	12-39	12-46#	13-19	13-80	13-166	13-188	13-251	13-271	13-312	13-325	13-350	13-376	13-426	13-449
	13-502	13-524	13-569	13-592	13-615	14-17	14-37	14-71	14-94	14-137	14-156	14-220	14-253	14-317
	14-341	14-395	14-415	14-468	14-490	14-535	14-564	14-627	14-645	14-693	14-723	14-787	15-1	
EXT1	5-638#													
EXT10	5-641#													
EXT2	5-639#													
EXT20	5-642#													
EXT4	5-640#													
EXT40	5-643#													
F1	5-551#													
F2	5-552#													
F3	5-553#													
F4	5-554#													
F5	5-555#													
FEN	5-664#													
FER	5-583#													
FMT22	5-700#	13-233	13-253											
GNS	6-1	6-1	11-6	11-29	15-1	15-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1
	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1
	26-1	26-1												
GO	5-550#													
GRV	5-563#													
GTSWR	11-29	26-1#												
HCE	5-586#													
HCI	5-698#													
HCRC	5-587#													
HT	5-501#	20-1	20-1											
IAE	5-589#													
IBSAVE	18-1	18-1	18-1	18-1	18-1	18-1#	18-1*	18-1*	18-1*					
IE	5-507#													
ILF	5-579#													
ILR	5-580#													
IOTVEC	5-501#	11-24*	11-24*											
IR	5-530#													
IXE	5-668#	5-686#												
KYBCTL	8-0#	12-54*	12-73*	13-19	13-19*	13-80	13-80*	13-166	13-166*	13-188	13-188*	13-251	13-251*	13-271
	13-271*	13-312	13-312*	13-325	13-325*	13-350	13-350*	13-376	13-376*	13-426	13-426*	13-449	13-449*	13-502
	13-502*	13-524	13-524*	13-569	13-569*	13-592	13-592*	13-615	13-615*	14-17	14-17*	14-37	14-37*	14-71
	14-71*	14-94	14-94*	14-137	14-137*	14-156	14-156*	14-220	14-220*	14-253	14-253*	14-317	14-317*	14-341
	14-341*	14-395	14-395*	14-415	14-415*	14-468	14-468*	14-490	14-490*	14-535	14-535*	14-564	14-564*	14-627
	14-627*	14-645	14-645*	14-693	14-693*	14-723	14-723*	14-787	14-787*	15-1				
LF	5-501#	20-1	20-1	27-8	27-9	27-11	27-12							
LST	5-570#													
MAXTN	12-67	31-29#												
MCLK	5-599#													
MCPE	5-512#	14-117	14-139	14-601	14-601	14-601	14-630	14-630	14-630					
MHS	5-666#	5-684#												
MINX	5-600#													
MOH	5-632#													
MOL	5-572#	13-46	13-46	13-147	13-147	13-168	13-168	13-233	13-234	13-234	13-253	13-257	13-257	13-299
	13-314	13-340	13-352	13-406	13-406	13-406	13-406	13-428	13-428	13-428	13-428	13-483	13-504	13-579
	13-579	13-594	13-594	13-644	13-644	14-19	14-19	14-54	14-54	14-73	14-73	14-117	14-117	14-117

ABCDEFGHIJKLMNOPQRSTUVWXYZ

	14-537	14-537	14-537	14-537	14-537	14-537	14-537	14-537	14-537	14-537	14-600	14-600	14-600	14-600
	14-600	14-600	14-600	14-600	14-600	14-600	14-600	14-600	14-600	14-600	14-600	14-629	14-629	14-629
	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629	14-629
	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-695	14-695	14-695	14-695
	14-695	14-765	14-765	14-765	14-765	14-765	14-765	14-765	14-765	14-765	14-789	14-789	14-789	14-789
	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789	14-789
PORTBI	12-27	27-7#												
PORTC	8-0#	12-30*	12-31*	12-32*	14-206	14-237								
PRO	5-501#													
PR1	5-501#													
PR2	5-501#													
PR3	5-501#													
PR4	5-501#													
PR5	5-501#	12-75	13-17											
PR6	5-501#	11-27												
PR7	5-501#													
PRE	5-716#													
PS	5-501	5-501#	12-47*	13-147*	13-147*	13-147*	13-147*	13-168*	13-168*	13-168*				
PSEL	5-511#													
PSU	5-713#													
PSW	5-501#													
PTNBR	8-0#	13-31*	13-31*	13-38*	13-38*	13-46*	13-46*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*	13-147*
	13-147*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-168*	13-233*	13-233*	13-233*	13-234*	13-234*
	13-234*	13-234*	13-234*	13-253*	13-253*	13-253*	13-257*	13-257*	13-257*	13-257*	13-257*	13-257*	13-299*	13-299*
	13-299*	13-299*	13-299*	13-299*	13-299*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-314*	13-340*
	13-340*	13-340*	13-340*	13-340*	13-352*	13-352*	13-352*	13-352*	13-352*	13-406*	13-406*	13-406*	13-406*	13-406*
	13-406*	13-406*	13-406*	13-406*	13-406*	13-406*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*	13-428*
	13-428*	13-428*	13-428*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-483*	13-504*	13-504*
	13-504*	13-504*	13-504*	13-504*	13-504*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-555*	13-579*
	13-579*	13-579*	13-579*	13-579*	13-579*	13-579*	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-594*	13-644*
	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	13-644*	14-19*	14-19*
	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-19*	14-54*	14-54*	14-54*	14-54*
	14-54*	14-54*	14-54*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-73*	14-117*	14-117*
	14-117*	14-117*	14-117*	14-117*	14-117*	14-117*	14-117*	14-117*	14-117*	14-117*	14-139*	14-139*	14-139*	14-139*
	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-139*	14-206*	14-206*	14-206*	14-206*	14-206*
	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*	14-206*
	14-206*	14-237*	14-237*	14-237*	14-237*	14-237*	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-302*	14-319*
	14-319*	14-319*	14-319*	14-319*	14-319*	14-319*	14-374*	14-374*	14-374*	14-374*	14-374*	14-374*	14-374*	14-397*
	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-397*	14-451*	14-451*	14-451*	14-451*	14-451*	14-451*	14-470*
	14-470*	14-470*	14-470*	14-518*	14-518*	14-518*	14-518*	14-518*	14-518*	14-518*	14-518*	14-518*	14-518*	14-537*
	14-537*	14-537*	14-537*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*	14-600*
	14-600*	14-600*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*	14-629*
	14-629*	14-679*	14-679*	14-679*	14-679*	14-679*	14-679*	14-679*	14-679*	14-695*	14-695*	14-695*	14-695*	14-765*
	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-765*	14-789*	14-789*	14-789*	14-789*	14-789*
	14-789*	29-1	29-3	29-5	29-6	29-7	29-11							
PWRVEC	5-501#													
R6	5-501#	11-24	11-24*	11-24*										
R7	5-501#													
RAW	5-679#													
RDCHR	23-1	26-1#												
RDLIN	24-1	26-1#												
RDOCT	12-12	12-62	12-88	26-1#										
RDY	5-508#													
RELERR	8-0#	13-147*	13-147*	13-168*	13-168*	13-234*	13-234*	13-257*	13-257*	13-299	13-299*	13-299*	13-314	13-314*
	13-314*	13-340*	13-340*	13-352*	13-352*	13-406*	13-406*	13-428*	13-428*	13-483*	13-483*	13-504*	13-504*	13-555
	13-555*	13-555*	13-579*	13-579*	13-594*	13-594*	13-594*	13-644*	13-644*	14-19*	14-19*	14-19*	14-54*	14-54*

SSCMRE	6-44#	7-0												
SSCMTM	6-44#	7-0	7-0	7-0	7-0	7-0								
SSESCA	5-501#													
SSNEWT	5-501#	13-19	13-80	13-166	13-188	13-251	13-271	13-312	13-325	13-350	13-376	13-426	13-449	13-502
	13-524	13-569	13-592	13-615	14-17	14-37	14-71	14-94	14-137	14-156	14-220	14-253	14-317	14-341
	14-395	14-415	14-468	14-490	14-535	14-564	14-627	14-645	14-693	14-723	14-787	14-794		
SSSET	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1	26-1#	
SSSKIP	5-501#													
.SACT1	5-490#	6-5												
.SCATC	5-491#	6-1												
.SCMTA	5-491#	6-44												
.SEOP	5-491#	15-1												
.SERRO	5-491#	18-1												
.SERRT	5-490#	19-1												
.SRDOC	5-492#	24-1												
.SREAD	5-492#	23-1												
.SSAVE	5-492#	25-1												
.SSCOP	5-491#	17-1												
.STRAP	5-492#	26-1												
.STYPD	5-492#	22-1												
.STYPE	5-491#	20-1												
.STYPO	5-492#	21-1												
.EQUAT	5-490#	5-501												
.HEADE	5-490#	5-498												
.SETUP	5-490#	10-192												
.SWRHI	5-490#	5-499												
.SWRLO	5-490#	5-499#												
AA	13-82#	13-147	13-168											
BB	13-190#	13-233	13-253											
BB1	13-212#	13-234	13-257											
CC	13-273#	13-299	13-314											
CC1	13-327#	13-340	13-352											
CHECK	5-93#	13-31	13-31	13-38	13-38	13-46	13-46	13-46	13-46	13-233	13-233	13-233	13-234	13-253
	13-253	13-253	13-257	13-299	13-299	13-314	13-314	13-406	13-406	13-406	13-406	13-428	13-428	13-428
	13-428	13-483	13-483	13-483	13-483	13-504	13-504	13-504	13-504	13-555	13-555	13-555	13-555	13-644
	13-644	14-19	14-19	14-54	14-73	14-117	14-139	14-206	14-206	14-302	14-302	14-302	14-302	14-319
	14-319	14-319	14-319	14-374	14-374	14-374	14-397	14-397	14-397	14-518	14-518	14-537	14-537	14-601
	14-601	14-601	14-630	14-630	14-630	14-765	14-765	14-789	14-789	14-189				
CLRATA	5-382#	13-579	13-594	13-644	14-19	14-54	14-73	14-117	14-139	14-206	14-237	14-302	14-302	14-319
	14-319	14-374	14-397	14-451	14-470	14-518	14-537	14-600	14-629	14-679	14-695	14-765	14-789	
COMMEN	5-501#													
DD	13-571#	13-579	13-594											
EE	13-617#	13-644	14-19											
ENDCOM	5-501#													
ERPOR	5-501#	12-97	13-31	13-31	13-38	13-38	13-46	13-46	13-46	13-46	13-56	13-147	13-147	13-147
	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147	13-147
	13-147	13-147	13-147	13-147	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168
	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-168	13-233	13-233	13-233
	13-234	13-234	13-234	13-234	13-234	13-234	13-253	13-253	13-253	13-253	13-257	13-257	13-257	13-257
	13-257	13-299	13-299	13-299	13-299	13-299	13-299	13-299	13-299	13-314	13-314	13-314	13-314	13-314
	13-314	13-340	13-340	13-340	13-340	13-352	13-352	13-352	13-352	13-406	13-406	13-406	13-406	13-406
	13-406	13-406	13-406	13-406	13-406	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428	13-428
	13-428	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-483	13-504	13-504	13-504	13-504	13-504
	13-504	13-504	13-504	13-555	13-555	13-555	13-555	13-555	13-555	13-555	13-555	13-579	13-579	13-579
	13-579	13-579	13-579	13-594	13-594	13-594	13-594	13-594	13-594	13-644	13-644	13-644	13-644	13-644
	13-644	13-644	13-644	14-19	14-19	14-19	14-19	14-19	14-19	14-19	14-19	14-54	14-54	14-54

	14-54	14-54	14-54	14-54	14-73	14-73	14-73	14-73	14-73	14-73	14-73	14-117	14-117	14-117
	14-117	14-117	14-117	14-117	14-117	14-117	14-117	14-139	14-139	14-139	14-139	14-139	14-139	14-139
	14-139	14-139	14-139	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-206	14-206
	14-206	14-206	14-206	14-206	14-206	14-237	14-237	14-237	14-237	14-302	14-302	14-302	14-302	14-302
	14-302	14-302	14-302	14-302	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-319	14-319
	14-374	14-374	14-374	14-374	14-374	14-374	14-397	14-397	14-397	14-397	14-397	14-397	14-397	14-397
	14-451	14-451	14-451	14-451	14-451	14-470	14-470	14-470	14-470	14-470	14-470	14-470	14-470	14-470
	14-518	14-518	14-518	14-537	14-537	14-537	14-537	14-537	14-537	14-600	14-600	14-600	14-600	14-600
	14-600	14-601	14-601	14-601	14-601	14-601	14-601	14-629	14-629	14-629	14-629	14-629	14-629	14-629
	14-630	14-630	14-630	14-630	14-630	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-679	14-679
	14-695	14-765	14-765	14-765	14-765	14-765	14-765	14-789	14-789	14-789	14-789	14-789	14-789	17-1
ESCAPE	5-501#													
FF	14-39#	14-54	14-73											
FF1	14-96#	14-117	14-139											
GETPRI	5-501#													
GETSWR	5-501#	11-29	11-29#											
GG	14-158#	14-206												
HEAD	5-444#	13-19	13-80	13-166	13-188	13-251	13-271	13-312	13-325	13-350	13-376	13-426	13-449	13-502
	13-524	13-569	13-592	13-615	14-17	14-37	14-71	14-94	14-137	14-156	14-220	14-253	14-317	14-341
	14-395	14-415	14-468	14-490	14-535	14-564	14-627	14-645	14-693	14-723	14-787			
HH	14-222#	14-237												
II	14-255#	14-302	14-319											
JJ	14-343#	14-374	14-397											
KK	14-417#	14-451	14-470											
LL	14-566#	14-600	14-629											
LL1	14-589#	14-601	14-630											
MM	14-492#	14-518	14-537											
MMO	13-526#	13-555												
MM1	13-451#	13-483	13-504											
MORETA	6-7#	7-0												
MSG	13-3#	13-19	13-62#	13-80	13-148#	13-166	13-170#	13-188	13-235#	13-251	13-259#	13-271	13-300#	13-312
	13-316#	13-325	13-341#	13-350	13-357#	13-376	13-407#	13-426	13-430#	13-449	13-484#	13-502	13-510#	13-524
	13-557#	13-569	13-580#	13-592	13-596#	13-615	13-645#	14-17	14-21#	14-37	14-55#	14-71	14-75#	14-94
	14-118#	14-137	14-141#	14-156	14-208#	14-220	14-239#	14-253	14-303#	14-317	14-321#	14-341	14-375#	14-395
	14-399#	14-415	14-452#	14-468	14-473#	14-490	14-5 9#	14-535	14-539#	14-564	14-602#	14-627	14-632#	14-645
	14-680#	14-693	14-702#	14-723	14-766#	14-787	14-792#							
MULT	5-501#													
NEUTRA	5-236#	13-147	13-168	13-234	13-257	13-299	13-314	13-340	13-352	13-406	13-428	13-483	13-504	13-555
	13-579	13-594	13-644	14-19	14-54	14-73	14-117	14-139	14-206	14-206	14-237	14-302	14-319	14-374
	14-397	14-451	14-470	14-518	14-537	14-600	14-629	14-679	14-695	14-765	14-789			
NEWTST	5-501#	13-19	13-80	13-166	13-188	13-251	13-271	13-312	13-325	13-350	13-376	13-426	13-449	13-502
	13-524	13-569	13-592	13-615	14-17	14-37	14-71	14-94	14-137	14-156	14-220	14-253	14-317	14-341
	14-395	14-415	14-468	14-490	14-535	14-564	14-627	14-645	14-693	14-723	14-787	14-794		
NN	14-725#	14-765	14-789											
OO	13-378#	13-406	13-428											
POP	5-501#	22-1	24-1	25-1										
PUSH	5-501#	22-1	24-1	25-1										
RELEAS	5-167#	13-299	13-314	13-406	13-428	13-483	13-504	13-579	13-594	13-644	13-644	14-19	14-19	14-54
	14-73	14-117	14-117	14-139	14-139	14-206	14-206	14-206	14-206	14-374	14-397	14-518	14-537	14-600
	14-600	14-629	14-629	14-679	14-695	14-765	14-789							
REPORT	5-501#													
RR	14-647#	14-679	14-695											
SEIZE	5-2#	13-147	13-168	13-299	13-314	13-406	13-428	13-483	13-504	13-579	13-594	13-644	14-19	14-54
	14-73	14-117	14-139	14-302	14-319	14-374	14-397	14-451	14-470	14-600	14-629	14-765	14-789	
SELECT	5-210#	13-31	13-31	13-38	13-38	13-46	13-46	13-147	13-147	13-147	13-168	13-168	13-168	13-233
	13-233	13-233	13-234	13-253	13-253	13-253	13-257	13-299	13-299	13-299	13-299	13-314	13-314	13-314

