

# RP04

DUAL CONTROLLER I  
MD-11-DZRPP-A

EP-DZRPP-A-DL-A

MAR 1976

COPYRIGHT ©1976

**digital**

FICHE 1 OF 1

Made In U.S.A.

DZRPPA SEQ



CO1

IDENTIFICATION

SEG 0001

PRODUCT CODE: MAINDEC-11-DZRPP-A-D

PRODUCT NAME: RPD: DUAL CONTROLLER LOGIC TEST - PART 1

DATE CREATED: DECEMBER 21, 1974

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: C. HESS

COPYRIGHT (C) 1974, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THE INFORMATION IN THIS STATEMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

ACTUAL DISTRIBUTION OF THE SOFTWARE DESCRIBED IN THIS DOCUMENT WILL BE SUBJECT TO TERMS AND CONDITIONS TO BE ANNOUNCED ON SOME FUTURE DATE BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

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

1. ABSTRACT
2. REQUIREMENTS
  - 2.1 EQUIPMENT
  - 2.2 PRELIMINARY PROGRAMS
  - 2.3 OTHER PROGRAMS
3. LOADING PROCEDURES
4. STARTING PROCEDURES
  - 4.1 STARTING ADDRESSES
  - 4.2 UNIBUS & VECTOR ADDRESSES
  - 4.3 OPERATOR ACTION
5. OPERATING PROCEDURES
  - 5.1 OPERATIONAL SWITCH SETTINGS
  - 5.2 TEST SELECTION
  - 5.3 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
- B. TEST DESCRIPTION
9. PROGRAM LISTING

1. ABSTRACT

THE RPO4 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RPO4 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 RPO4 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.

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 TEST BY THIS PROGRAM.

2. REQUIREMENTS

## 2.1 EQUIPMENT

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

## 2.2 PRELIMINARY PROGRAMS

RPO4 DISKLESS CONTROLLER TEST  
PART 1 (MAINDEC-11-DZRPS)  
PART 2 (MAINDEC-11-DZRPT)

RPO4 FUNCTIONAL CONTROLLER TEST  
PART 1 (MAINDEC-11-DZRPV)  
PART 2 (MAINDEC-11-DZRPV)

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

## 2.3 OTHER PROGRAMS

- A. THE OPERATION OF THE UNLOAD COMMAND AND THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH ARE TESTED BY THE RPO4 DUAL CONTROLLER LOGIC TEST, PART 2 (MAINDEC-11-DZRPQ).
- B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RPO4 MULTIDRIVE EXERCISER (MAINDEC-11-DZRPN-B). NOTE THAT THE RPO4 EXERCISER MUST BE PROGRAM REVISION 'B' OR LATER. REVISION 'A' OF THE RPO4 EXERCISER DOES NOT SUPPORT DUAL CONTROLLER OPERATION.

3. LOADING PROCEDURES

THE PROGRAM MAY BE LOADED BY THE ABSOLUTE PAPER TAPE LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE MEDIA USING THE ASSOCIATED 'XXDP' LOADED. THE PROGRAM MAY BE INCLUDED IN AN 'XXDP' CHAIN.

4. STARTING PROCEDURES4.1 STARTING ADDRESSES

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

4.2 UNIBUS & VECTOR ADDRESSES

THE PROGRAM ASSURES THE FOLLOWING UNIBUS AND VECTOR ADDRESSES. THESE ADDRESSES MAY BE CHANGED PRIOR TO INITIATING A PROGRAM START AT ANY OF THE STARTING LOCATIONS.

<u>MEMORY LOCATION</u>	<u>CONTENTS</u>	<u>FUNCTION</u>
1136	177560	TTY KEYBOARD STATUS REG
1140	177562	TTY KEYBOARD BUFFER REG
1142	177564	TTY PRINTER STATUS REG
1144	177566	TTY PRINTER BUFFER REG
1204	172540	KH11-P STATUS REG
1206	172542	KH11-L COUNTER BUFFER
1210	104	KH11-P VECTOR ADDRESS
1212	177546	KH11-L STATUS REGISTER
1214	100	KH11-L VECTOR ADDRESS
1270	176700	RH11/RP04 ADDRESS
1272	254	RH11 INTERRUPT VECTOR ADDRESS

4.3 OPERATOR ACTION

- A. CONNECT THE DUAL CONTROLLER TEST CABLE BETWEEN BUS A & BUS B ON THE RP04 BEING TESTED. (SEE SECTION 5.3)
- B. LOAD THE PROGRAM INTO MEMORY IN THE PROPER PROCESSOR.
- C. SWITCH THE 'CONTROLLER SELECT' SWITCH ON THE RP04 TO BE TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
- D. LOAD THE APPROPRIATE STARTING ADDRESS (200(8) OR 210(8)) INTO THE SWITCH REGISTER.
- E. PRESS START.
- F. ENTER THE DRIVE NUMBER. (THIS MUST BE THE NUMBER DISPLAYED IN THE DRIVE NUMBER LED ON THE CONTROL

- PANEL.)  
 G. ENTER THE NUMBER OF THE TEST TO BE RUN.  
 H. THE PROGRAM CAN BE STOPPED AT ANY TIME AND RESTARTED FROM LOCATION 204.

## 5. OPERATING PROCEDURES

-----

### 5.1 OPERATIONAL SWITCH SETTINGS

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

THE SWITCH SETTINGS ARE:

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

### 5.2 TEST SELECTION

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

THE 'RUBOUT KEY' (RO) CAN BE USED TO DELETE THE LAST CHARACTER ENTERED. SUCCESSIVE STUCKING AT 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 THE ENTIRE ENTRY BY TYPING A 'CONTROL U' (↑U).

### 5.3 TEST CABLE CONNECTION

TO TEST THE RPO4 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 RPO4 BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE TEST CABLE CONNECTED TO THE RPO4 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT OF THE RPO4 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.)

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 DEVICE ON THE SYSTEM (RPO4 OR NON-RPO4 DEVICE) WHICH \*  
 \* HAS ANY ADDRESS IN CONFLICT WITH EITHER OF THE TEST \*  
 \* ADDRESSES MUST BE TURNED OFF. \*  
 \*\*\*\*\*

THE TEST CABLE CONNECTION TO THE RPO4 UNDER TEST WILL DEPEND ON WHICH PROCESSOR/RH11 IS TO TEST THE RPO4. IF THE RPO4 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 RPO4 IS TO BE TESTED BY PROCESSOR ON PORT B, 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 'RHAS' (ATTENTION SUMMARY REGISTER) IS READ. THE ATTENTION BIT POSITION IS DETERMINED BY THE SWITCH ON THE 'DP' BOARD (M7775). THE ATTENTION BIT THAT APPEARS FOR THE DRIVE IS THE INCLUSIVE 'OR' OF THE PORT A & PORT B ATTENTION BITS. BECAUSE OF THIS, THE PROGRAM LOOKS AT ONLY THE ATTENTION BIT IN 'RHDS1' (DRIVE STATUS REGISTER) TO DETERMINE THE STATE OF THE SELECTED PORT'S ATTENTION BIT.

## 6. ERRORS

WHEN THE PROGRAM ENCOUNTERS AN ERROR, THE ERROR ROUTINE IS CALLED AND THE 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 RPO4 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 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 19AP 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 ONE-SHOT VALUE MEASURED THROUGH EACH PORT.
- B. TEST 4 AND TEST 5. THESE TESTS SET 'VV-A' AND 'VV-B', RESPECTIVELY. THESE TESTS MUST BE PERFORMED AT LEAST ONCE BEFORE TESTS 6 - 46 ARE RUN.

### 7.7 DISK SURFACE USAGE

THE DIAGNOSTIC DOES NOT USE THE DISK SURFACE. HOWEVER, THE

DRIVE MUST BE CYCLED UP AND 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
06	4000
07	4000
10	100
11	100
12	4000
13	4
14	4
15	4
16	4000
17	4000
20	4000
21	4000
22	4000
23	4000
24	4000
25	4000
26	4000
27	4000
28	4000
29	4
30	4
31	4000
32	4000
33	4000
34	4
35	4
36	4
37	4
40	4
41	4
42	4
43	4000
44	4000
45	4000
46	4000

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

TESTS 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 OCCURRING. BECAUSE THE PROGRAM MUST RESET THE RPO4 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR WAS DETECTED

## 8. TEST DESCRIPTION

-----

### 8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

THE PROGRAM DETERMINES IF AN RPO4 IS IN NEUTRAL BY CHECKING THE CONTENTS OF THE DRIVE STATUS REGISTER (RHDS1) 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 RHDS1, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DISCREPANCY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

### 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 (RHDS1) THROUGH THE SEIZING PORT AND VERIFYING THAT CORRECT STATUS IS SEEN. WHEN RHDS1 IS READ THROUGH THE OPPOSITE PORT, ZEROS SHOULD BE SEEN. IF BOTH CONDITIONS EXIST, (I.E., CORRECT STATUS THROUGH THE SEIZING PORT AND ZEROS THROUGH THE OPPOSITE PORT), THE PROGRAM CONCLUDES THAT THE DRIVE HAS BEEN SEIZED BY THE SPECIFIED PORT.

### 8.3 TEST 1 - DRIVE ACCESS TEST

VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS

A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RHDS1 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 RHDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.

B. READ EACH DRIVE REGISTER, EXCEPT RHCS1, 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.

B.5 TEST 3 - PORT 'B' SEIZE/TIMEOUT TEST

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

- A. WRITE 0'S INTO RHDS1 THROUGH PORT 'B'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. READ EACH DRIVE REGISTER, EXCEPT RHCS1, 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 TIME SHOT AND SAVE THE VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO NEUTRAL.

B.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.

B.7 TEST 5 - PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

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

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

B.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 RHDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- B.9 TEST 7 - TEST RELEASE, DRIVE SEIZED BY PORT 'B'
- TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.
- B.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.
- B.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.
- B.12 TEST 12 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
- VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR DOES NOT CAUSE THE SEIZING PORT TO RELEASE THE DRIVE.
- A. SEIZE THE DRIVE BY WRITING 0'S INTO RHDS1 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 PORT 'A' 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 NEITHER ATTENTION BIT IS SET.
- B.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 RHDS1 THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

- B.13. B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

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

- A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
- C. ISSUE A MASSBUS CLEAR.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

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

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

- A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
- C. ISSUE A MASSBUS CLEAR.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

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

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

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

B.17 TEST 17 - TEST SEIZE BY RHCS1 READ THROUGH PORT 'A'

VERIFY THAT READING THE CONTROL REGISTER (RHCS1) SEIZES THE DRIVE.

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

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

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

- A. READ THE CONTROL REGISTER (RHCSI) 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 RHDS1.
- B. WRITE 0'S INTO RHDS1 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 RHDS1.
- B. WRITE 0'S INTO RHDS1 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 RHCSI THROUGH PORT 'A'

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

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

B.22 TEST 24 - TEST NO 'PORT REQUEST' WHEN READ RHCSI THROUGH PORT 'B'

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

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

B.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 RHCSI.
- 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.

B.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 RHCSI.
- 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 (RHAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER PORT.

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

8.26 TEST 30 - TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT

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

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

8.27 TEST 31 - TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
- B. WRITE 1'S INTO RHER1 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 RHDS1.
- B. WRITE 1'S INTO RHER1 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 RHDS1.
- B. WRITE 1'S INTO RHER1 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 RHER1 HAS NOT BEEN CLEARED.
- D. CLEAR RHER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
- B. WRITE 1'S INTO RHER1 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 RHER1 HAS NOT BEEN CLEARED.
- D. CLEAR RHER1 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 RHDS1.
- B. WAIT 500 MS AND WRITE 0'S INTO RHDS1 THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN  $\pm 25\%$  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.

8.32 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 RHDS1.
- B. WAIT 500 MS AND WRITE 0'B INTO RHDS1 THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN  $\pm 25\%$  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.

SEA 0017

## 8.33 TEST 37 - 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 RHDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RHDS1 FROM PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT BEEN RELEASED.
- E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

## 8.34 TEST 40 - 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 RHDS1.
- B. SET PORT REQUEST BY WRITING 0'S INTO RHDS1 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.

## 8.35 TEST 41 - 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.

## 8.36 TEST 42 - 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.

8.37 TEST 43 - 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 RHDS1.
- B. WRITE 1'S INTO RHER1, RHER2, & RHER3 THROUGH PORT 'A'.
- C. READ RHER1, RHER2, & RHER3 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RHER1, RHER2, & RHER3 THROUGH PORT 'A'.
- E. WRITE 1'S INTO RHER1, RHER2, & RHER3 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.

8.38 TEST 44 - 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 RHDS1.
- B. WRITE 1'S INTO RHER1, RHER2, & RHER3 THROUGH PORT 'B'.
- C. READ RHER1, RHER2, & RHER3 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RHER1, RHER2, & RHER3 THROUGH PORT 'B'.
- E. WRITE 1'S INTO RHER1, RHER2, & RHER3 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.

- 8.39 TEST 45 - 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 RHDS1.
  - C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

- 8.40 TEST 46 - 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 RHDS1.
  - C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

9. PROGRAM LISTING

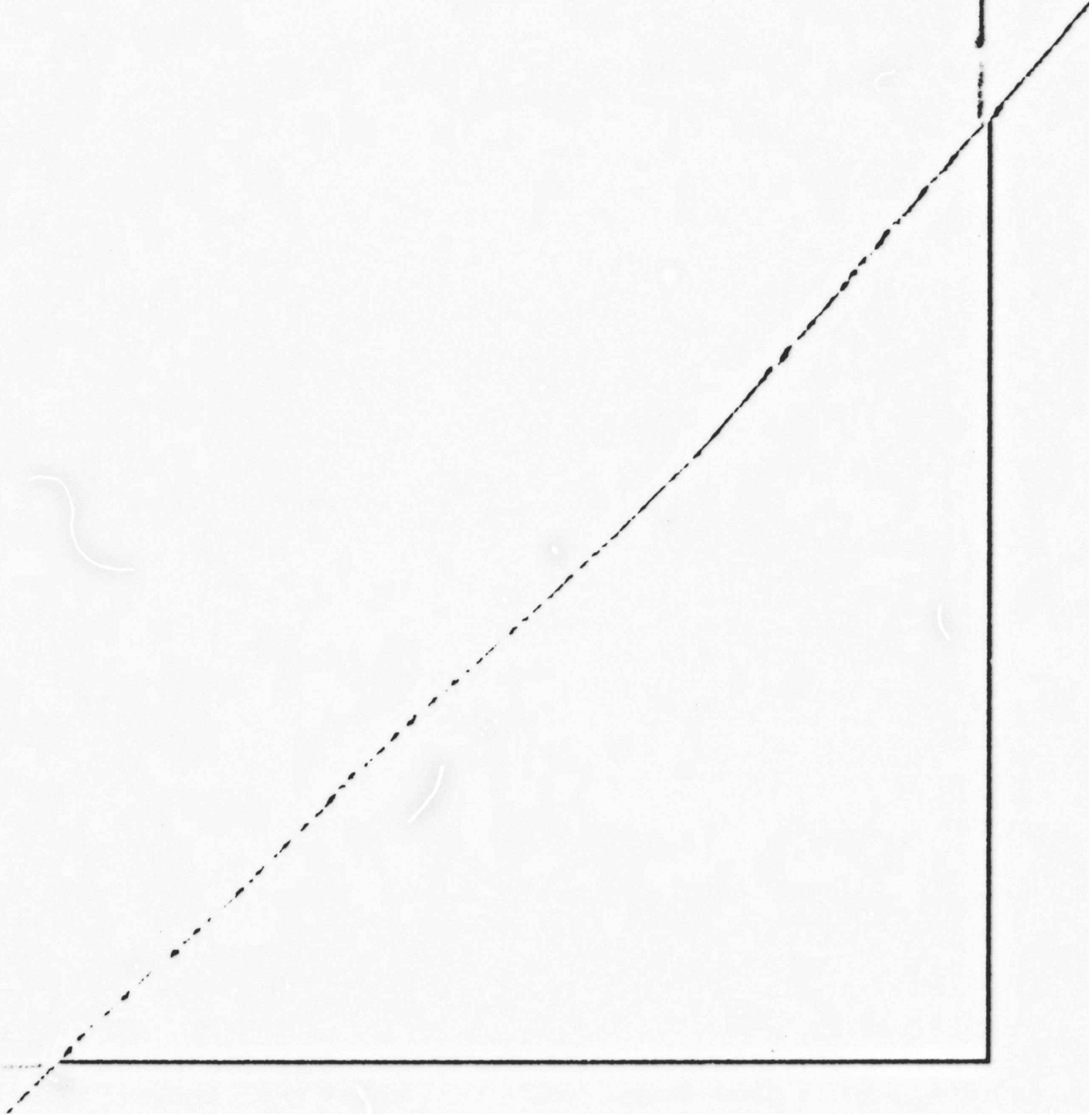
4245	OPERATIONAL SWITCH SETTINGS
4248	BASIC DEFINITIONS
4252	RH11 REGISTERS
4287	RPO4 REGISTERS
4447	DEFINITIONS OF THE RH11/RPO4 ADDRESS INDEXES
4472	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
4521	COMMON TAGS
(3)	RH11/RPO4 UNIBUS AND VECTOR ADDRESSES
(1)	ERROR POINTER TABLE
4851	STARTUP AND INITIALIZATION ROUTINES
4952	*** TESTS ***
4971	T1 DRIVE ACCESS TEST
5036	T2 PORT 'A' SEIZE/TIMEOUT TEST
5129	T3 PORT 'B' SEIZE/TIMEOUT TEST
5150	T4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
5220	T5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'
5240	T6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'
5285	T7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'
5298	T10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
5326	T11 TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
5351	T12 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
5407	T13 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
5430	T14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
5491	T15 TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
5513	T16 TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
5564	T17 TEST SEIZE BY RHCSI READ THROUGH PORT 'A'
5590	T20 TEST SEIZE BY RHCSI READ THROUGH PORT 'B'
5614	T21 TEST 'PORT REQUEST' FROM PORT 'A'
5668	T22 TEST PORT REQUEST FROM PORT 'B'
5688	T23 TEST NO 'PORT REQUEST' WHEN READ RHCSI THROUGH PORT 'A'
5726	T24 TEST NO 'PORT REQUEST' WHEN READ RHCSI THROUGH PORT 'B'
5749	T25 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
5797	T26 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
5816	T27 TEST SEIZE BY WRITING ATTENTION BIT
5886	T30 TEST NO SEIZE WHEN 'D' WRITTEN INTO ATTENTION BIT
5922	T31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
5995	T32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
6019	T33 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
6079	T34 TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
6099	T35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
6158	T36 TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
6183	T37 TEST NO TIMEOUT THROUGH PORT 'A'
6255	T40 TEST NO TIMEOUT THROUGH PORT 'B'
6276	T41 TEST PORT 'A' ATTENTION AFTER A COMMAND
6327	T42 TEST PORT 'B' ATTENTION AFTER A COMMAND
6356	T43 TEST PORT INTERACTION FROM PORT 'A'
6420	T44 TEST PORT INTERACTION FROM PORT 'B'
6438	T45 TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
6494	T46 TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
6501	*** SUBROUTINES ***
6556	'SYSMAC' UTILITY ROUTINES
6566	END OF PASS ROUTINE
6570	SCOPE HANDLER ROUTINE

## TABLE OF CONTENTS

6571	ERROR HANDLER ROUTINE
6572	ERROR MESSAGE TIMEOUT ROUTINE
6573	TYPE ROUTINE
6574	BINARY TO OCTAL (ASCII) AND TYPE
6575	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6576	TTY INPUT ROUTINE
6577	READ AN OCTAL NUMBER FROM THE TTY
6578	SAVE AND RESTORE RD-RS ROUTINES
6579	TRAP DECODER
(3)	TRAP TABLE
6583	TELETYPE MESSAGES
6603	TEST ERROR MESSAGES
6757	CONSTANTS, TABLES, ETC

K02

3665







(1)	000100	PR2=	100	::PRIORITY LEVEL	2
(1)	000140	PR3=	140	::PRIORITY LEVEL	3
(1)	000200	PR4=	200	::PRIORITY LEVEL	4
(1)	000240	PR5=	240	::PRIORITY LEVEL	5
(1)	000300	PR6=	300	::PRIORITY LEVEL	6
(1)	000340	PR7=	340	::PRIORITY LEVEL	7

(1)		:*"SWITCH REGISTER" SWITCH DEFINITIONS			
(1)	100000	SW15=	100000		
(1)	040000	SW14=	40000		
(1)	020000	SW13=	20000		
(1)	010000	SW12=	10000		
(1)	004000	SW11=	4000		
(1)	002000	SW10=	2000		
(1)	001000	SW09=	1000		
(1)	000400	SW08=	400		
(1)	000200	SW07=	200		
(1)	000100	SW06=	100		
(1)	000040	SW05=	40		
(1)	000020	SW04=	20		
(1)	000010	SW03=	10		
(1)	000004	SW02=	4		
(1)	000002	SW01=	2		
(1)	000001	SW00=	1		

(1)		.EQUIV	SW09, SW9
(1)		.EQUIV	SW08, SW8
(1)		.EQUIV	SW07, SW7
(1)		.EQUIV	SW06, SW6
(1)		.EQUIV	SW05, SW5
(1)		.EQUIV	SW04, SW4
(1)		.EQUIV	SW03, SW3
(1)		.EQUIV	SW02, SW2
(1)		.EQUIV	SW01, SW1
(1)		.EQUIV	SW00, SW0

(1)		:*DATA BIT DEFINITIONS (BIT00 TO BIT15)			
(1)	100000	BIT15=	100000		
(1)	040000	BIT14=	40000		
(1)	020000	BIT13=	20000		
(1)	010000	BIT12=	10000		
(1)	004000	BIT11=	4000		
(1)	002000	BIT10=	2000		
(1)	001000	BIT09=	1000		
(1)	000400	BIT08=	400		
(1)	000200	BIT07=	200		
(1)	000100	BIT06=	100		
(1)	000040	BIT05=	40		
(1)	000020	BIT04=	20		
(1)	000010	BIT03=	10		
(1)	000004	BIT02=	4		
(1)	000002	BIT01=	2		
(1)	000001	BIT00=	1		

(1)		.EQUIV	BIT09, BIT9
(1)		.EQUIV	BIT08, BIT8

```

(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0

(1) ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1) 000004 ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
(1) 000010 RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
(1) 000014 TBITVEC=14 ;: "T" BIT
(1) 000014 TRTVEC= 14 ;: TRACE TRAP
(1) 000014 BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 ;: POWER FAIL
(1) 000030 EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC=34 ;: "TRAP" TRAP
(1) 000060 TKVEC= 60 ;: TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 ;: TTY PRINTER VECTOR
(1) 000240 PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR

;*****

.SBTTL RH11 REGISTERS

;*****

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

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

;CONTROL AND STATUS REGISTER 2 (RHCS2)

4264 000001 US1= 1 ;: UNIT SELECT (BIT #0)
4265 000002 US2= 2 ;: UNIT SELECT (BIT #1)
4266 000004 US4= 4 ;: UNIT SELECT (BIT #2)
4267 000010 BAI= 10 ;: BUS ADDRESS INCREMENT INHIBIT (BIT #3)
4268 000020 PAT= 20 ;: MASSBUS PARITY TEST (BIT #4)
4269 000040 CLR= 40 ;: CLEAR (BIT #5)
4270 000100 IR= 100 ;: INPUT READY (BIT #6)
4271 000200 OR= 200 ;: OUTPUT READY (BIT #7)
4272 000400 MPE= 400 ;: MASS BUS PARITY ERROR (BIT #8)
4273 001000 MXF= 1000 ;: MISSED TRANSFER ERROR (BIT #9)
4274 002000 PGE= 2000 ;: PROGRAM ERROR (BIT #10)
4275 004000 NEM= 4000 ;: NON EXISTANT MEMORY (BIT #11)
4276 010000 NED= 10000 ;: NON EXISTANT DRIVE (BIT #12)
4277 020000 UPE= 20000 ;: UNIBUS PARITY ERROR (BIT #13)
4278 040000 WCE= 40000 ;: WRITE CHECK ERROR (BIT #14)
4279 100000 DLT= 100000 ;: DATA LATE (BIT #15)
4280

```

```

4281 ;DATA BUFFER REGISTER (RHDB)
4282 ;EACH BIT IS CALLED BY BIT NUMBER
4283
4284
4285 ;*****
4286
4287 .SBTTL RPO4 REGISTERS
4288
4289 ;*****
4290
4291 ;CONTROL AND STATUS 1 REGISTER. (#00)
4292
4293 GO= 1 ;GO (BIT #0)
4294 IE= 100 ;INTERRUPT ENABLE (BIT #6)
4295 RDY= 200 ;READY (BIT #7)
4296 A16= 400 ;HIGH ORDER UNIBUS BITS (BIT #8)
4297 A17= 1000 ;HIGH ORDER UNIBUS BITS (BIT #9)
4298 PSEL= 2000 ;PORT SELECT (BIT #10)
4299 DVA= 4000 ;DEVICE AVAILABLE BIT #11)
4300 MCPE= 20000 ;MASSBUS PARITY ERROR (BIT #13)
4301 TRE= 40000 ;TRANSFER ERROR (BIT #14)
4302 SC= 100000 ;SPECIAL CONDITION (BIT #15)
4303
4304 ;STATUS REGISTER (RHDS1) (#01)
4305
4306 ;DFS= 1 DRIVE FORWARD 5"/SEC. (BIT #0)
4307 DFF20= 2 ;DRIVE FORWARD 2"/SEC. (BIT #1)
4308 DICB= 4 ;DRIVE TO INNER GUARD BAND (BIT #2)
4309 GKV= 10 ;GO REVERSE (BIT #3)
4310 DL64= 20 ;DIFFERENCE LESS THAN 64 (BIT #4)
4311 DE1= 40 ;DIFFERENCE EQUALS 1 (BIT #5)
4312 VV= 100 ;VOLUME VALID (BIT #6)
4313 DRY= 200 ;DRIVE READY (BIT #7)
4314 DPR= 400 ;DRIVE PRESENT (BIT #8)
4315 PGM= 1000 ;PROGRAMMABLE (BIT #9)
4316 LST= 2000 ;LAST SECTOR TRANSFERRED (BIT #10)
4317 WRL= 4000 ;WRITE LOCK (BIT #11)
4318 MOL= 10000 ;MEDIUM ON-LINE (BIT #12)
4319 PIP= 20000 ;POSITIONING OPERATION IN PROGRESS (BIT #13)
4320 ERR= 40000 ;COMPOSITE ERROR (BIT #14)
4321 ATA= 100000 ;ATTENTION ACTIVE (BIT #15)
4322
4323 ;ERROR REGISTER #01 (RHER1) (#02)
4324
4325 ;ILF= 1 ;ILLEGAL FUNCTION (BIT #0)
4326 ILR= 2 ;ILLEGAL REGISTER (BIT #1)
4327 PMR= 4 ;REGISTER MODIFICATION REFUSED (BIT #2)
4328 PAR= 10 ;PARITY ERROR (BIT #3)
4329 FER= 20 ;FORMAT ERROR (BIT #4)
4330 WCF= 40 ;WRITE LOCK FAIL (BIT #5)
4331 ECH= 100 ;ECC HARD ERROR (BIT #6)
4332 HCE= 200 ;HEADER COMPARE ERROR (BIT #7)
4333 HCRC= 400 ;HEADER CRC ERROR (BIT #8)
4334 AOE= 1000 ;ADDRESS OVERFLOW ERROR (BIT #9)
    
```

4335 002000  
4336 004000  
4337 010000  
4338 020000  
4339 040000  
4340 100000

IAE= 2000  
WLE= 4000  
DTE= 10000  
OPI= 20000  
UNS= 40000  
DCK= 100000

;INVALID ADDRESS ERROR (BIT #10)  
;WRITE LOCK ERROR (BIT #11)  
;DRIVE TIMING ERROR (BIT #12)  
;OPERATION INCOMPLETE (BIT #13)  
;DRIVE UNSAFE (BIT #14)  
;DATA CHECK ERROR (BIT 15)

;MAINTAINABILITY REGISTER (RHMR) (#33)

4341 000001  
4342 000002  
4343 000004  
4344 000010  
4345 000020  
4346 000040  
4347 000200

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

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

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

4348 000001  
4349 000002  
4350 000004  
4351 000010  
4352 000020  
4353 000040  
4354 000100  
4355 000200

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

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

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

;DRIVE TYPE REGISTER (RHDT) (#06)  
;EACH BIT IS CALLED BY BIT NUMBER

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

4356 000001  
4357 000002  
4358 000004  
4359 000010  
4360 000020  
4361 000040  
4362 000100  
4363 000200  
4364 000400  
4365 001000  
4366 002000  
4367 004000  
4368 010000  
4369 020000  
4370 040000  
4371 100000

EXT1= 1  
EXT2= 2  
EXT4= 4  
EXT10= 10  
EXT20= 20  
EXT40= 40  
SC1= 100  
SC2= 200  
SC4= 400  
SC10= 1000  
SC20= 2000  
TRK1= 4000  
TRK2= 10000  
TRK4= 20000  
TRK10= 40000  
TRK20= 100000

;EXTENSION 1 (BIT #0)  
;EXTENSION 2 (BIT #1)  
;EXTENSION 3 (BIT #2)  
;EXTENSION 4 (BIT #3)  
;EXTENSION 5 (BIT #4)  
;EXTENSION 6 (BIT #5)  
;SECTOR COUNT FIELD 0 (BIT #6)  
;SECTOR COUNT FIELD 1 (BIT #7)  
;SECTOR COUNT FIELD 2 (BIT #8)  
;SECTOR COUNT FIELD 3 (BIT #9)  
;SECTOR COUNT FIELD 4 (BIT #10)  
;TRACK FIELD 1 (BIT #11)  
;TRACK FIELD 2 (BIT #12)  
;TRACK FIELD 3 (BIT #13)  
;TRACK FIELD 4 (BIT #14)  
;TRACK FIELD 5 (BIT #15)

;ERROR REGISTER #2 (RHER2) (#10)

4388

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

000001  
000002  
000004  
000010  
000020  
000040  
000100  
000200  
000400  
001000  
002000  
004000  
010000  
020000  
100000  
  
000001  
000002  
000004  
000010  
000020  
000040  
000200  
002000  
004000  
010000  
  
000001  
000032  
000010  
000020  
000040  
000100  
040000  
100000

WS=  
CSU=  
MSE=  
TDF=  
TUF=  
FEN=  
WRU=  
MHS=  
NHS=  
IXE=  
VU30=  
PLU=  
ACU=

: OFFSET REGISTER (RHOF)

OF25=  
OF50=  
OF100=  
OF200=  
OF400=  
OF800=  
OFREV=  
HCI=  
ECI=  
FMT22=

: DESIRED CYLINDER ADDRESS (RHCA) (#12)  
: EACH BIT IS CALLED BY BIT NUMBER.

: CURRENT CYLINDER ADDRESS (RHCC) (#13)  
: EACH BIT IS CALLED BY BIT NUMBER.

: SERIAL NUMBER REGISTER (RHSN) (#14)  
: EACH BIT IS CALLED BY BIT NUMBER.

: ERROR REGISTER #03 (RHER3) (#15)

PSU=  
VUF=  
UNP=  
PRE=  
ACL=  
DCL=  
SKT=  
CYL=

: ECC POSITION AND REGISTER (RHEC1) (#16)  
: EACH BIT IS CALLED BY BIT NUMBER.

: ECC PATTERN REGISTER (RHEP2) (#17)

: WRITE CURRENT UNSAFE (BIT #0)  
: CURRENT SINK FAILURE (BIT #1)  
: WRITE SELECT UNSAFE (BIT #2)  
: CURRENT SWITCH UNSAFE (BIT #3)  
: MOTOR SEQUENCE ERROR (BIT #4)  
: TRANSITIONS DETECTOR FAILURE (BIT #5)  
: TRANSITIONS UNSAFE (BIT #6)  
: FAILSAFE ENABLED (BIT #7)  
: WRITE READY UNSAFE (BIT #8)  
: MULTIPLE HEAD SELECT (BIT #9)  
: NO HEAD SELECTION (BIT #10)  
: INDEX ERROR (BIT #11)  
: 30VOLT UNSAFE (BIT #12)  
: PLO UNSAFE (BIT #13)  
: AC UNSAFE (BIT #15)

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

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

4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
4473  
4474  
4475  
4476  
4477  
4478  
4479  
4480  
4481  
4520

000000  
000002  
000004  
000006  
000010  
000012  
000014  
000016  
000020  
000022  
000024  
000026  
000030  
000032  
000034  
000036  
000040  
000042  
000044  
000046

:EACH BIT IS CALLED BY BIT NUMBER

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

.SRTL DEFINITIONS OF THE RH11/RPO4 ADDRESS INDEXES

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

RHCS=0  
RHMC=2  
RHBA=4  
RHDA=6  
RHGA=10  
RHHA=12  
RHIA=14  
RHJA=16  
RHKA=20  
RHLA=22  
RHMA=24  
RHNA=26  
RHOA=30  
RHPA=32  
RHQA=34  
RHRA=36  
RHOA=40  
RHPA=42  
RHOA=44  
RHOA=46

:CONTROL REGISTER #1 (DRIVE REG. 00)  
:WORD COLLECTOR REGISTER (NOT A DRIVE REG)  
:UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)  
:DESIRED SERIAL NUMBER ADDRESS REGISTER (DRIVE REG. 05)  
:CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)  
:DRIVE STATUS REGISTER #1 (DRIVE REG. 01)  
:ERROR REGISTER #1 (DRIVE REG. 02)  
:ATTENTION REGISTER #1 (DRIVE REG. 04)  
:LOOK AHEAD REGISTER (DRIVE REG. 03)  
:DATA BUFFER REGISTER (NOT A DRIVE REG.)  
:MAINTAINABILITY REGISTER (DRIVE REG. 08)  
:DRIVE TYPE REGISTER (DRIVE REG. 06)  
:SERIAL NUMBER REGISTER (DRIVE REG. 10)  
:OFFSET REGISTER (DRIVE REG. 11)  
:DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)  
:CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)  
:ERROR REGISTER #2 (DRIVE REG. 14)  
:ERROR REGISTER #3 (DRIVE REG. 15)  
:ECC POSITION REGISTER (DRIVE REG. 16)  
:ECC PATTERN REGISTER (DRIVE REG. 17)

.SRTL TRAP CATCHER

:#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "02" HALT  
:#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS  
:#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

.SRTL STARTING ADDRESS(ES)  
.=200

000200 000137 002044  
000204 000137 002410  
000210 000137 002360

JMP @START ; JUMP TO STARTING ADDRESS OF PROGRAM  
:#STARTING ADDRESS IS LOCATION 200  
JMP EXEC ; RESTART  
:#RESTART ADDRESS IS LOCATION 204  
JMP CHANGE ; CHANGE RH11 ADDRESS AND START  
:#START AT LOCATION 210 TO CHANGE THE RH11 ADDRESS FROM 176700

```

4521 ;*****
(1) .SBTTL COMMON TAGS
(1) ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1) ;*USED IN THE PROGRAM.
(1) 000046 000046 .=46
(1) 000046 057626 SENDAD ;;LOGICAL END OF PROGRAM
(1) 001100 001100 .=1100
(1) 001100 SCMTAG:
(1) 001100 000000 SPASS: .WORD 0
(1) 001102 000 STSTNM: .BYTE 00
(1) 001103 000 SERFLG: .BYTE 00
(1) 001104 000000 SICNT: .WORD 00
(1) 001106 000000 SLPADR: .WORD 00
(1) 001110 000000 SLPERR: .WORD 00
(1) 001112 000000 SERTTL: .WORD 00
(1) 001114 000 SITEMB: .BYTE 00
(1) 001115 001 SERMAX: .BYTE 01
(1) 001116 000000 SERAPC: .WORD 00
(1) 001120 000000 SGOADR: .WORD 00
(1) 001122 000000 SBDADR: .WORD 00
(1) 001124 000000 SGO DAT: .WORD 00
(1) 001126 000000 SBD DAT: .WORD 00
(1) 001130 000000 000000 000000 SBDAT: .WORD 0,0,0
(1) 001136 177560 STKS: 177560
(1) 001140 177562 STKB: 177562
(1) 001142 177564 STPS: 177564
(1) 001144 177566 STPB: 177566
(1) 001146 000 SNULL: .BYTE 0
(1) 001147 002 SFILLS: .BYTE 2
(1) 001150 012 SFILLC: .BYTE 12
(1) 001151 000 STPFLG: .BYTE 0
(1) 001152 000000 SREGAD: .WORD 0
(1) 001154 000000 SREGO: .WORD 0
(1) 001156 000000 STMP1: .WORD 00
(1) 001160 000000 STMP2: .WORD 00
(1) 001162 000000 STMP3: .WORD 00
(1) 001164 000000 STMP4: .WORD 00
(1) 001170 000000 STINES: 0
(1) 001174 177607 000377 SBELL: .ASCIZ <207><377><377>
(1) 001200 077 SQUES: .ASCII /?/
(1) 001201 015 SCRFLF: .ASCII <15>
(1) 001202 000012 SLF: .ASCIZ <12>
(3) 001204 172540 SLKCSR: .WORD 172540
(3) 001206 172542 SLKCSB: .WORD 172542
(3) 001210 000104 SLPVEC: .WORD 104
(3) 001212 177546 SLKS: .WORD 177546

```

```

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

```



(3)	001214	000100	\$LLVEC: .WORD	100	; ADDR OF KW11-L VECTOR
(3)	001216	000000	PORTA: .WORD	0	; ADDRESS OF PORT A
(3)	001220	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
(3)	001222	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
(3)	001224	000000	ASR1: .WORD	0	; ATA-A OR ATA-B = 1
(3)	001226	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
(3)	001230	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
(3)	001232	000000	OPPRT: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
(3)	001234	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
(3)	001236	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURED
(3)	001240	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
(3)	001242	000000	RELEA: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
(3)	001244	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
(3)	001246	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
(3)	001250	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
(3)	001252	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
(3)	001254	000000	TIMEAM: .WORD	0	; PORT A TIMEOUT VALUE - 25%
(3)	001256	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
(3)	001260	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
(3)	001262	000000	TIMEBM: .WORD	0	; PORT B TIME VALUE - 25%
(3)	001264	000000	TIMES: .WORD	0	; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
(3)	001266	000000	KYECTL: .WORD	0	; SINGLE TEST INDICATOR

(3) ;\*\*\*\*\*

(3) .SBTTL RH11/RPO4 UNIBUS AND VECTOR ADDRESSES

(3) ;\*\*\*\*\*

(3)	001270	176700	\$RPADR: .WORD	176700	; RH11/RPO4 UNIBUS ADDRESS
(3)	001272	000254	\$RPVEC: .WORD	254	; RH11 INTERRUPT VECTOR ADDRESS

(3)

(2)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558

```

;*****
.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
SERRTB:
;ERROR 1
            EM1          ;WRONG DRIVE TYPE
            DH1
            DT1
            DF1
;ERROR 2
            EM2          ;DRIVE NOT ON LINE
            DH1
            DT1
            DF1
;ERROR 3
            EM3          ;SERIAL NUMBERS NOT THE SAME
            DH3
            DT3
            DF1
;ERROR 4
            EM4          ;DRIVE NOT SEIZED BY PORT 'N'
            DH4
            DT7
            DF7
;ERROR 5
            EM5          ;WRONG STATUS SEEN BY THE SEIZING PORT
            DH5
            DT5
            DF5
;ERROR 6

```

```

001274
001274 062471
001276 066464
001300 070214
001302 070460
001304 062512
001306 066464
001310 070214
001312 070460
001314 062534
001316 066535
001320 070230
001322 070460
001324 062616
001326 066604
001330 070276
001332 070473
001334 062647
001336 066727
001340 070244
001342 070465

```

4559					
4560	001344	062715	EM6		;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA
4561	001346	067177	DH13		
4562	001350	070316	DT13		
4563	001352	070465	DF5		
4564					
4565				;ERROR 7	
4566					
4567	001354	063015	EM7		;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
4568	001356	067003	DH7		
4569	001360	070276	DT7		
4570	001362	070473	DF7		
4571					
4572				;ERROR 10	
4573					
4574	001364	063076	EM10		;REGISTER CONTENTS INCORRECT
4575	001366	066727	DH5		
4576	001370	070244	DT5		
4577	001372	070465	DF5		
4578					
4579				;ERROR 11	
4580					
4581	001374	063126	EM11		;CONTROL BUS PARITY ERROR WHILE READING REGISTER
4582	001376	067126	DH11		
4583	001400	070214	DT1		
4584	001402	070460	DF1		
4585					
4586				;ERROR 12	
4587					
4588	001404	063212	EM12		;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
4589	001406	067673	DH36		
4590	001410	070404	DT37		
4591	001412	070506	DF36		
4592					
4593				;ERROR 13	
4594					
4595	001414	063262	EM13		; 'VOLUME VALID' BIT NOT SET BY READIN PRESET
4596	001416	067177	DH13		
4597	001420	070316	DT13		
4598	001422	070465	DF5		
4599					
4600				;ERROR 14	
4601					
4602	001424	063347	EM14		; 'VOLUME VALID' SET ON THE OPPOSITE PORT
4603	001426	067177	DH13		
4604	001430	070316	DT13		
4605	001432	070465	DF5		
4606					
4607				;ERROR 15	
4608					
4609	001434	063412	EM15		;THE ATTI BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
4610	001436	067003	DH7		
4611	001440	070276	DT7		
4612	001442	070473	DF7		

4613				
4614				
4615				
4616	001444	063471	EM16	;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
4617	001446	067003	DH7	
4618	001450	070276	DT7	
4619	001452	070473	DF7	
4620				
4621				
4622				
4623	001454	063544	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
4624	001456	067003	DH7	
4625	001460	070276	DT7	
4626	001462	070473	DF7	
4627				
4628				
4629				
4630	001464	063623	EM20	;DRIVE NOT SEIZED - ATTN BIT FOR PORT CLEARED
4631	001466	067673	DH36	
4632	001470	070404	DT37	
4633	001472	070506	DF36	
4634				
4635				
4636				
4637	001474	063703	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
4638	001476	067673	DH36	
4639	001500	070404	DT37	

A.P11

## ERROR POINTER TABLE

4641	001502	070506	DF36	
4642				
4643				;ERROR 22
4644				
4645	001504	063756	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
4646	001506	067317	DH22	
4647	001510	070334	DT22	
4648	001512	070502	DF31	
4649				
4650				;ERROR 23
4651				
4652	001514	064043	EM23	;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
4653	001516	067415	DH23	
4654	001520	070346	DT23	
4655	001522	070460	DF1	
4656				
4657				;ERROR 24
4658				
4659	001524	064111	EM24	;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
4660	001526	067415	DH23	
4661	001530	070346	DT23	
4662	001532	070460	DF1	
4663				
4664				
4665				;ERROR 25
4666				
4667	001534	064170	EM25	;TIMEOUT ONE-SHOT DID NOT RETRIGGER
4668	001536	067673	DH36	
4669	001540	070374	DT36	
4670	001542	070506	DF36	
4671				
4672				
4673				;ERROR 26
4674				
4675	001544	064233	EM26	;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
4676	001546	067317	DH22	
4677	001550	070334	DT22	
4678	001552	070502	DF31	
4679				
4680				;ERROR 27
4681				
4682	001554	064320	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
4683	001556	067003	DH7	
4684	001560	070276	DT7	
4685	001562	070473	DF7	
4686				
4687				;ERROR 30
4688				
4689	001564	064376	EM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
4690	001566	067673	DH36	
4691	001570	070374	DT36	
4692	001572	070506	DF36	
4693				
4694				;ERROR 31

4695				
4696	001574	064473	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WIITH REQUEST SE
4697	001576	067574	DH31	
4698	001600	070362	DT31	
4699	001602	070502	DF31	
4700				
4701				;ERROR 32
4702				
4703	001604	064550	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
4704	001606	066727	DH5	
4705	001610	070244	DT5	
4706	001612	070465	DF5	
4707				
4708				;ERROR 33
4709				
4710	001614	064621	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
4711	001616	067673	DH36	
4712	001620	070374	DT36	
4713	001622	070506	DF36	
4714				
4715				;ERROR 34
4716				
4717	001624	064723	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
4718	001626	067673	DH36	
4719	001630	070374	DT36	
4720	001632	070506	DF36	
4721				
4722				;ERROR 35
4723				
4724	001634	055026	EM35	;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
4725	001636	067673	DH36	
4726	001640	070404	DT37	
4727	001642	070506	DF36	
4728				
4729				;ERROR 36
4730				
4731	001644	065105	EM36	;TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS
4732	001646	067673	DH36	
4733	001650	070374	DT36	
4734	001652	070506	DF36	
4735				
4736				;ERROR 37
4737				
4738	001654	065156	EM37	;DRIVE IS NON-EXISTANT
4739	001656	067673	DH36	
4740	001660	070404	DT37	
4741	001662	070506	DF36	
4742				
4743				;ERROR 40
4744				
4745	001664	065224	EM40	;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
4746	001666	066464	DH1	
4747	001670	070346	DT23	
4748	001672	070460	DF1	

4749				
4750			;ERROR 41	
4751				
4752	001674	065301	EM41	;TIMEOUT CLEARED ATTENTION BIT
4753	001676	067415	DH23	
4754	001700	070346	DT23	
4755	001702	070460	DF1	
4756				
4757			;ERROR 42	
4758				
4759	001704	065343	EM42	;DRIVE NOT IN NEUTRAL OR SEIZED
4760	001706	067722	DH42	
4761	001710	070414	DT42	
4762	001712	070511	DF42	
4763				
4764			;ERROR 43	
4765				
4766	001714	065431	EM43	;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
4767	001716	067722	DH42	
4768	001720	070414	DT42	
4769	001722	070511	DF42	
4770				
4771			;ERROR 44	
4772				
4773	001724	065506	EM44	;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
4774	001726	067741	DH44	
4775	001730	070362	DT31	
4776	001732	070502	DF31	
4777				
4778			;ERROR 45	
4779				
4780	001734	065563	EM45	;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
4781	001736	066464	DH1	
4782	001740	070214	DT1	
4783	001742	070460	DF1	
4784				
4785			;ERROR 46	
4786				
4787	001744	065642	EM46	;CAN'T ACCESS DRIVE THROUGH EITHER PORT
4788	001746	070037	DH46	
4789	001750	070422	DT46	
4790	001752	070502	DF31	
4791				
4792			;ERROR 47	
4793				
4794	001754	065711	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
4795	001756	067415	DH23	
4796	001760	070346	DT23	
4797	001762	070460	DF1	
4798				
4799			;ERROR 50	
4800				
4801	001764	065777	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
4802	001766	067177	DH13	

4803	001770	070316	DT13	
4804	001772	070465	DF5	
4805				
4806				;ERROR 51
4807				
4808	001774	066062	EM51	;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
4809	001776	066727	DH5	
4810	002000	070244	DT5	
4811	002002	070465	DF5	
4812				
4813				;ERROR 52
4814				
4815	002004	066145	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
4816	002006	067177	DH13	
4817	002010	070316	DT13	
4818	002012	070465	DF5	
4819				
4820				;ERROR 53
4821				
4822	002014	066243	EM53	;CAN'T READ ATTN BIT FROM OPPOSITE PORT
4823	002016	067415	DH23	
4824	002020	070214	DT1	
4825	002022	070460	DF1	
4826				
4827				;ERROR 54
4828				
4829	002024	066324	EM54	;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
4830	002026	067317	DH22	
4831	002030	070434	DT54	
4832	002032	070502	DF31	
4833				
4834				;ERROR 55
4835				
4836	002034	066417	EM55	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
4837	002036	070135	DH55	
4838	002040	070446	DT55	
4839	002042	070513	DF55	
4840				
4841				
4842				
4849				;*****
4850				.SBTTL STARTUP AND INITIALIZATION ROUTINES
4851				;*****
4852				
4853				
4854				
4855	002044			START:

```

(1) 002044 012737 000340 177776 MOV #340,2#PS ;: LOCK OUT ALL INTERRUPTS
(1) 002052 012706 001100 MOV #SCMTAG,R6 ;: FIRST LOCATION TO BE CLEARED
(1) 002056 005026 CLR (R6)+ ;: CLEAR MEMORY LOCATION
(1) 002060 022706 001136 CMP #STKS,R6 ;: DONE?
(1) 002064 001374 BNE .-6 ;: LOOP BACK IF NO
(1) 002066 012706 001100 MOV #STACK,SP ;: SETUP THE STACK POINTER
(1) 002072 012737 057662 000020 MOV #SSCOPE,2#IOTVEC ;: IOT VECTOR FOR SCOPE ROUTINE
    
```



```

(1) 002100 012737 000340 000022      MOV      #340, @IOTVEC+2 ;:LEVEL 7
(1) 002106 012737 060040 000030      MOV      #SEERR, @EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
(1) 002114 012737 000340 000032      MOV      #340, @EMTVEC+2 ;:LEVEL 7
(1) 002122 012737 061650 000034      MOV      #STRAP, @TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
(1) 002130 012737 000340 000036      MOV      #340, @TRAPVEC+2 ;:LEVEL 7
(1) 002136 013737 057556 057550      MOV      SENDCT, SEOPCT ;:SETUP END-OF-PROGRAM COUNTER
(1) 002144 005037 001170      CLR      $TIMES ;:INITIALIZE NUMBER OF ITERATIONS
(1) 002150 005037 001172      CLR      $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 002154 112737 000001 001115      MOV      #1, $ERMAX ;:ALLOW ONE ERROR PER TEST
(1) 002162 012737 002162 001106      MOV      #., $LPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 002170 012737 002170 001110      MOV      #., $LPERR ;:SETUP THE ERROR LOOP ADDRESS
4856 002176 000005      RESET ;:CLEAR THE SYSTEM
4857 002200 104400 061714      START1: TYPE ;:TITLE
4858 002204 012737 000240 002200      MOV      #NOP, START1 ;:TYPE PROGRAM NAME
4859 002212 012737 000240 002202      MOV      #NOP, START1+2 ;:DISABLE TITLE TYPEOUT AFTER INITIAL START
4860 002220 104400 062012      15: TYPE ;:FROM LOCATION 200 OR 210
4861 002224 104416      RDOCT ;:ENTER DRIVE ADDRESS
4862 002226 012637 001216      MOV      (SP)+, PORTA ;:GET THE ADDRESS
4863 002232 023727 001216 000007      CMP      PORTA, #7 ;:STORE THE ADDRESS
4864 002240 101403      BLOS    25 ;:SEE IF ADDRESS TOO LARGE
4865 002242 104400 062042      TYPE ;:BR IF NOT
4866 002246 000764      BR ;:TYPE ADDRESS ERROR MESSAGE
4867 002250 013737 001216 001220 25: MOV      PORTA, PORTB ;:TRY AGAIN
4868 002256 005237 001220      INC      PORTB ;:GENERATE THE PORT B ADDRESS
4869 002262 042737 000006 001220      BIC      #6, PORTB ;:INCREMENT THE ADDRESS
4870 002270 013746 001216      MOV      PORTA, -(SP) ;:LEAVE BIT 0
4871 002274 042716 177771      BIC      #1C6, (SP) ;:PUT PORT A ADDRESS ON THE STACK
4872 002300 052637 001220      BIS      (SP)+, PORTB ;:SAVE BITS 1 & 2
4873 002304 104400 062064      TYPE ;:SET BITS 1 & 2 IN PORT B ADDRESS
4874 002310 013746 001216      MOV      PORTA IS ;:PORT A ADDRESS IS
4875 002314 104410      TYPDS ;:PUT THE ADDRESS ON THE STACK
4876 002316 104400 062112      TYPE ;:TYPE PORT A ADDRESS
4877 002322 013746 001220      MOV      PORTB IS ;:PORT B ADDRESS IS
4878 002326 104410      TYPDS ;:PUT ADDRESS ON THE STACK
4879 002330 104400 001201      TYPE ;:TYPE PORT B ADDRESS
4880 002334 013737 001216 001222      MOV      $CR LF ;:ANOTHER CR-LF
4881 002342 062737 000006 001222      MOV      PORTA, PORTC ;:GENERATE ADDRESS OF DRIVE NOT TESTED
4882 002350 042737 177770 001222      ADD      #6, PORTC ;:COMPLEMENT SOME BITS
4883 002356 013701 001216      BIC      #1C7, PORTC ;:SAVE ONLY LOWER BITS
4884 002362 116137 070636 001224      MOV      PORTA, R1 ;:USE PORT A ADDRESS AS INDEX
4885 002370 004737 057264      MOV      ATABIT(R1), ASR1 ;:GET ATTENTION BIT FOR DRIVE
4886 002374 000137 002410      JSR      PC, CKCLK ;:SETUP CLOCK
4887 002400 104400 062140      JMP      EXEC ;:CLOCK HAS BEEN STARTED
4888 002404 000000      TYPE ;:NO CLOCK ON SYSTEM
4889 002406 000776      HALT ;:FATAL ERROR
4890      BR      .-2 ;:INTERLOCK HALT
;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
4891
4892
4893 002410 000005      EXEC: RESET ;:CLEAR EVERYTHING
4894 002412 012737 000240 177776      MOV      #(<5*32.>), PS ;:SET PROCESSOR PRIORITY TO 5
4895 002420 104400 001201      TYPE ;:CR-LF
4896 002424 013700 001270      MOV      $RPADR, R0 ;:RH11 ADDRESS FOR INDEXING
4897 002430 012706 001100      MOV      #STACK, R6 ;:LOAD STACK POINTER
4898 002434 004737 057264      JSR      PC, CKCLK ;:START THE CLOCK
    
```

```

4899 002440 000240      NOP      ;RETURN IF NO CLOCK
4900 002442 005037 001266  CLR      KYBCTL ;CLEAR SINGLE TEST INDICATOR
4901 002446 005037 001100  CLR      $PASS  ;CLEAR THE PASS COUNT
4902 002452 112737 000001 001115  MOVVB   #1,$ERMAX ;SET ERROR MAX TO 1
4903 002460 012737 002460 001106  MOV     #.,$LPADR ;INITIAL SETTING FOR LOOP ADDRESS
4904 002466 012737 002466 001110  MOV     #.,$LPERR ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
4905 002474 104400 062207      TYPE    ,TESTNO  ;ASK FOR TEST NUMBER
4906 002500 104416      RDOCT   ;GET THE NUMBER
4907 002502 012601      MOV     (SP)+,R1 ;PUT ENTRY INTO R1
4908 002504 001002      BNE     .+6      ;BR IF NOT ZERO
4909 002506 000137 002710  JMP     TST1    ;ENTER ZERO - PERFORM ALL TESTS
4910 002512 020137 070646  CMP     R1,MAXTN ;SEE IF NUMBER GREATER THAN MAXIMUM
4911 002516 003403      BLE     1$     ;BR IF LESS OR EQUAL
4912 002520 104400 062227      TYPE    ,BADNO   ;BAD ENTRY
4913 002524 000731      BR      EXEC   ;TRY AGAIN
4914 002526 005301      1$:    DEC    R1 ;DECREMENT ENTRY
4915 002530 006301      ASL    R1     ;SHIFT IT LEFT
4916 002532 016137 070520 002556  MOV     TSTADR(R1),2$ ;GET THE TEST ADDRESS
4917 002540 005237 001266      INC    KYBCTL  ;SET SINGLE TEST INDICATOR
4918 002544 012737 000001 001104  MOV     #1,$ICNT ;PRESET ITERATION COUNT
4919 002552 000177 000000      JMP    2$     ;GO TO THE SELECTED TEST
4920 002556 000000      2$:    .WORD  0 ;TEST ADDRESS GOES HERE
4921
4922      ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
4923
4924 002560 000005      CHANGE: RESET ;CLEAR THE SYSTEM
4925 002562 012737 000340 177776  MOV     #340,$APS ;LOCK OUT ALL INTERRUPTS
4926 002570 012706 001100      MOV     $STACK,$SP ;LOAD THE STACK POINTER
4927 002574 012737 061650 000034  MOV     $TRAP,$TRAPVEC ;LOAD TRAP VECTOR
4928 002602 012737 000340 000036  MOV     #340,$TRAPVEC+2 ;LEVEL 7
4929 002610 104400 062267      TYPE    ,ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS
4930 002614 013746 001270      MOV     $RPADR,-(SP) ;PUT THE ADDRESS ON THE STACK
4931 002620 104402      TYPOC   ;TYPE THE ACTUAL ADDRESS
4932 002622 104400 001201      TYPE    ,$CRLF  ;CR-LF
4933 002626 104400 062347      TYPE    ,NTRH11 ;ASK FOR NEW ADDRESS
4934 002632 104416      RDOCT   ;
4935 002634 005716      TST     (SP)   ;0 OR 'CR' ENTERED ?
4936 002636 001402      BEQ     1$     ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
4937 002640 011637 001270      MOV     (SP),$RPADR ;NEW RH11 ADDRESS
4938 002644 012737 002666 000004 1$:    MOV     #2$,$ ;LOAD TRAP ADDRESS
4939 002652 013700 001270      MOV     $RPADR,R0 ;RH11 ADDRESS
4940 002656 062700 000002      ADD    #2,R0   ;FORM ADDRESS OF RHW
4941 002662 005710      TST    (R0)   ;SEE IF RH11 RESPONDS AT THAT ADDRESS
4942 002664 000405      BR     3$     ;BR, RH11 ALIVE AT PRESENT ADDRESS
4943 002666 104400 062401      2$:    TYPE    ,NORESP ;REPORT NO RESPONSE
4944 002672 010046      MOV     R0,-(SP) ;SETUP TO CONVERT THE ADDRESS
4945 002674 104402      TYPOC   ;TYPE THE ADDRESS
4946 002676 000730      BR     CHANGE ;GET ADDRESS AGAIN
4947 002700 000137 002044      3$:    JMP    START  ;GO TO THE STARTING ADDRESS
4948
4949
4950      ;;*****
4951
4952      .SBTTL  *** TESTS ***

```

```

4953
4954
4955
4968
4969 002704 013700 001270
4970
4971
(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3) 002710
(3) 002710 000004
(3) 002712 005737 001266
(3) 002716 001406
(3) 002720 100002
(3) 002722 000137 002410
(3) 002726 012737 177777 001266
(3) 002734 112737 000001 001102
(3) 002742 012737 002764 001106
(3) 002750 012737 002764 001110
(1) 002756 012737 000001 001170
4972
4973
(2)
(1)
(1)
(1) 002764
4974
4975
4976
4984 002764 113760 001216 000010
(2) 002772 013737 001216 001226
(1) 003000 005760 000012
(2) 003004 005037 001236
(2) 003010 016037 000010 001126
(2) 003016 012737 000010 001122
(2) 003024 060037 001122
(2) 003030 005037 001124
(2) 003034 013737 001126 001156
(2) 003042 042737 167777 001156
(2) 003050 023737 001124 001156
(2) 003056 001414
(2) 003060 013737 001126 001166
(2) 003066 042737 010000 001166
(2) 003074 053737 001166 001124

```

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

TST1AA: MOV SRPADR,RO ;;RESTORE RO AFTER END OF PASS

\*\*\*\*\*

\*TEST 1 DRIVE ACCESS TEST

\*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS

\* A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RHDS1 HAS 'MOL' 'PGM' 'DPR' & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.

\* B. THE TEST IS REPEATED THROUGH BOTH PORTS.

\*\*\*\*\*

TST1:

```

SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #1,$TSTNM ;TEST NUMBER
MOV #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST1,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #1,$TIMES ;;DO 1 ITERATION

```

;;\*\*\*\*\*  
;END OF 'SCOPE' SETUP - START OF MAIN TEST

TEST1:

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

```

MOVB PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RHDS1(RO) ;SEE IF DRIVE (PORT A) PRESENT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHCS2(RO),SBDDAT ;GET CONTENTS OF RHCS2
MOV #RHCS2,SBOPADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBOPADR ;ADD RH11 BASE ADDRESS
CLR SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMPO ;MOVE REGISTER CONTENTS TO 'TMPO'
BIC #ICNED,$TMPO ;SAVE SPECIFIED BITS
CMP SGDDAT,$TMPO ;COMPARE THE BITS
BEQ 64$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #NED,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

```

(2) 003100 104037          ERROR 37          ;TYPE MESSAGE 37
(2) 003104 005137 001236      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 003108 000240          NOP
(1) 003112 005737 001236      64$:  TST      CKERR      ;WAS 'NED' SET ?
(1) 003116 001403          BEQ      .+10      ;BR IF NOT
(1) 003120 012760 000040 000010  MOV      #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
(2) 003126 011376 001220 000010  MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(2) 003134 013737 001220 001226  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 003142 005760 000012          TST      RHDS1(RO)   ;SEE IF DRIVE (PORT B) PRESENT
(2) 003146 005037 001236      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 003152 016037 000010 001126  MOV      RHCS2(RO),SBDDAT ;GET CONTENTS OF RHCS2
(2) 003160 012737 000010 001122  MOV      #RHCS2,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 003166 060037 001126          ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 003172 005037 001126          CLR      SGDDAT      ;WHAT REGISTER SHOULD BE
(2) 003176 013737 001126 001156  MOV      SBDDAT,STMPD   ;MOVE REGISTER CONTENTS TO 'STMPD'
(2) 003204 042737 167777 001155  BIC      #1CNED,STMPD  ;SAVE SPECIFIED BITS
(2) 003212 023737 001124 001156  CMP      SGDDAT,STMPD  ;COMPARE THE BITS
(2) 003220 001414          BEQ      65$        ;BR IF OK
(2) 003222 013737 001126 001166  MOV      SBDDAT,STMP4  ;COPY 'BAD DATA'
(2) 003230 042737 010000 001166  STC      #NED,STMP4   ;CLEAR THE MASKED BITS
(2) 003236 053737 001166 001124  MOV      STMP4,SGDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 003244 104037          ERROR 37          ;TYPE MESSAGE 37
(2) 003246 005137 001236      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 003252 000240          NOP
(1) 003254 005737 001236      65$:  TST      CKERR      ;WAS 'NED' SET ?
(1) 003260 001403          BEQ      .+10      ;BR IF NOT
(1) 003262 012760 000040 000010  MOV      #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'

```

```

4985
4986
4987
4988
4992 *****
;CONFIRM THAT DRIVE IS AN RPO4 AND IS DUAL PORT

```

```

(2) 003270 113760 001216 000010  MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 003276 013737 001216 001226  MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 003304 005037 001236          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 003310 016037 000026 001126  MOV      RHDT(RO),SBDDAT ;GET CONTENTS OF RHDT
(2) 003316 012737 000026 001122  MOV      #RHDT,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 003324 060037 001122          ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 003330 012737 024020 001124  MOV      #24020,SGDDAT ;WHAT REGISTER SHOULD BE
(2) 003336 023737 001124 001126  CMP      SGDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 003344 001403          BEQ      66$        ;BR IF OK
(2) 003346 104001          ERROR 1          ;TYPE MESSAGE 1
(2) 003350 005137 001236      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 003354 000240          NOP
(2) 003356 113760 001220 000010  66$:  MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(2) 003364 013737 001220 001226  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 003372 005037 001236          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 003376 016037 000026 001126  MOV      RHDT(RO),SBDDAT ;GET CONTENTS OF RHDT
(2) 003404 012737 000026 001122  MOV      #RHDT,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 003412 060037 001122          ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 003416 012737 024020 001124  MOV      #24020,SGDDAT ;WHAT REGISTER SHOULD BE
(2) 003424 023737 001124 001126  CMP      SGDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 003432 001403          BEQ      67$        ;BR IF OK
(2) 003434 104001          ERROR 1          ;TYPE MESSAGE 1
(2) 003436 005137 001236      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR

```

```

(2) 003442 000240
4993
4994
4995
4996
5001 003444 113760 001216 000010
(2) 003452 013737 001216 001226
(2) 003460 005037 001236
(2) 003464 016037 000012 001126
(2) 003472 012737 000012 001122
(2) 003500 060037 001122
(2) 003504 012737 001000 001124
(2) 003512 013737 001126 001156
(2) 003520 042737 176777 001156
(2) 003526 023737 001124 001156
(2) 003534 001414
(2) 003536 013737 001126 001166
(2) 003544 042737 001000 001166
(2) 003552 053737 001166 001124
(2) 003560 104045
(2) 003562 005137 001236
(2) 003566 000240
(2) 003570 005037 001236
(2) 003574 016037 000012 001126
(2) 003602 012737 000012 001122
(2) 003610 060037 001122
(2) 003614 012737 010600 001124
(2) 003622 012737 001126 001156
(2) 003630 042737 167177 001156
(2) 003636 023737 001124 001156
(2) 003644 001414
(2) 003646 013737 001126 001166
(2) 003654 042737 010600 001166
(2) 003662 053737 001166 001124
(2) 003670 104002
(2) 003672 005137 001236
(2) 003676 000240
(2) 003700 113760 001220 000010
(2) 003706 013737 001220 001226
(2) 003714 005037 001236
(2) 003720 016037 000012 001126
(2) 003726 012737 000012 001122
(2) 003734 060037 001122
(2) 003740 012737 001000 001124
(2) 003746 013737 001126 001156
(2) 003752 042737 176777 001156
(2) 003758 023737 001124 001156
(2) 003764 001414
(2) 003770 013737 001126 001166
(2) 004000 012737 001000 001166
(2) 004006 012737 001166 001124
(2) 004014 0045
(2) 004016 001236
(2) 004022 0000

```

```

675: NOP
:*****
:VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
MOV B PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBDADR ;ADD RH11 BASE ADDRESS
MOV #PGM,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CPGM,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 685 ;BR IF OK
MOV SBDAT,STMP4 ;COPY 'BAD DATA'
BIC #PGM,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 45 ;TYPE MESSAGE 45
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
685:
NOP
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBDADR ;ADD RH11 BASE ADDRESS
MOV #MOL!DPR!DRY,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #10600,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 695 ;BR IF OK
MOV SBDAT,STMP4 ;COPY 'BAD DATA'
BIC #10600,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 2 ;TYPE MESSAGE 2
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
695:
NOP
MOV B PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBDADR ;ADD RH11 BASE ADDRESS
MOV #PGM,SGDDAT ;WHAT REGISTER SHOULD BE
MOV SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CPGM,STMP0 ;SAVE SPECIFIED BITS
CMP SGDDAT,STMP0 ;COMPARE THE BITS
BEQ 705 ;BR IF OK
MOV SBDAT,STMP4 ;COPY 'BAD DATA'
BIC #PGM,STMP4 ;CLEAR THE MASKED BITS
BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 45 ;TYPE MESSAGE 45
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
705:
NOP

```

```

(2) 004024 005037 001236          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 004030 016037 000012 J01126    MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 004036 012737 000012 001122    MOV      #RHDS1, $BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 004044 060037 001122          ADD      RO, $BDAADR    ;ADD RHI1 BASE ADDRESS
(2) 004050 012737 010600 001124    MOV      #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 004056 013737 001126 001156    MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 004064 042737 167177 001156    BIC      #1C10600, $TMP0 ;SAVE SPECIFIED BITS
(2) 004072 023737 001124 001156    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
(2) 004100 001414          BEQ      71$            ;BR IF OK
(2) 004102 013737 001126 001166    MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
(2) 004110 042737 010600 001166    BIC      #10600, $TMP4   ;CLEAR THE MASKED BITS
(2) 004116 053737 001166 001124    BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 004124 104002          ERROR   2              ;TYPE MESSAGE 2
(2) 004126 005137 001236          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 004132 000240          NOP

```

71\$:

```

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

```

```

5002
5003
5004
5005
5006 004134 113760 001216 000010    MOV      PORTA, RHCS2(RO) ;SELECT PORT A
5007 004142 016037 000030 001124    MOV      RHN(RO), $GDDAT ;STORE THE PORT A SERIAL NUMBER
5008 004150 113760 001220 000010    MOV      PORTB, RHCS2(RO) ;SELECT PORT B
5009 004156 016037 000030 001126    MOV      RHN(RO), $BDDAT ;STORE THE PORT B SERIAL NUMBER
5010 004164 023737 001124 001126    CMP      $GDDAT, $BDDAT  ;ARE THEY THE SAME ?
5011 004172 001406          BEQ      1$            ;BR IF THEY ARE
5012 004174 104003          ERROR   3              ;REPORT THE ERROR
5013 004176 032737 100000 177570    BIT      #SW15, SWR      ;HALT ON ERROR ?
5014 004204 001001          BNE      1$            ;BR IF SET - PROGRAM HAS ALREADY HALTED
5015 004206 000000          HALT

```

1\$:

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(1)
(1)
(1)
(1) 004210 105737 001103          TSTB    SERFLG          ;DID AN ERROR OCCUR ?
(3) 004214 001412          BEQ      TST2            ;BR IF NOT
(1) 004216 032737 001000 177570    BIT      #SW09, SWR      ;SEE IF LOOP ON ERROR SET (SWR9=1)
(3) 004224 001406          BEQ      TST2            ;BR IF NOT
(1) 004226 105037 001103          CLRB    SERFLG          ;CLEAR THE ERROR FLAG
(1) 004232 005037 001170          CLR     $TIMES           ;CLEAR THE MAX ITERATION COUNT
(1) 004236 000177 174646          JMP     $SLPERR         ;GO TO THE LOOP ADDRESS

```

5017  
5035  
5036

```

*****
*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 RHDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE
*      HAS BEEN SEIZED.
*
*  B.  READ EACH DRIVE REGISTER, EXCEPT RHCS1, THROUGH PORT 'B',
*      VERIF. THAT 0'S ARE READ FROM EACH REGISTER.
*

```

(4) ;\* C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.  
(4) ;\* MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE  
(4) ;\* VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO  
(4) ;\* NEUTRAL.  
(4) ;\*  
(3) ;\*\*\*\*\*

(3) 004242  
(3) 004242 000004  
(3) 004244 005737 001266  
(3) 004250 001406  
(3) 004252 100002  
(3) 004254 000137 002410  
(3) 004260 012737 177777 001266 15:  
(3) 004266 112737 000002 001102 25:  
(3) 004274 012737 004316 001106  
(3) 004302 012737 004316 001110  
(1) 004310 012737 000012 001170  
5037  
5110

TEST2:  
SCOPE ;INITIALIZE THE SCOPE HANDLER  
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?  
BEQ 25 ;BR IF NOT  
BPL 15 ;BR IF JUST ENTERED TEST  
JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
15: MOV @-1, KYBCTL ;SET SINGLE TEST INDICATOR  
25: MOVB @2, \$STNM ;TEST NUMBER  
MOV @TEST2, \$LPADR ;LOAD LOOP ON TEST ADDRESS  
MOV @TEST2, \$LPERR ;LOAD LOOP ON ERROR ADDRESS  
MOV @10., \$TIMES ;DO J. ITERATIONS

;;\*\*\*\*\*  
;END OF 'SCOPE' SETUP - START OF MAIN TEST

(2) 004316  
(1) 004316 012737 000240 177776  
(1) 004324 005037 001250  
(1) 004330 005037 001252  
(1) 004334 005037 001254  
(2)  
(3)  
(2)  
(2)

TEST2:  
MOV @(&5\*32.), @PS ;SET PRIORITY TO 5 IN CASE LOOPING  
CLR TIMEA ;CLEAR TIMEOUT VALUE FOR PORT A  
CLR TIMEAP ;CLEAR UPPER TIMEOUT TOLERANCE  
CLR TIMEAM ;CLEAR LOWER TIMEOUT TOLERANCE

;;\*\*\*\*\*  
;START THE TIMER

(2) 004340 005037 001244  
(2) 004344 012737 003720 001246  
(1)  
(2)  
(2)

CLR TIME ;CLEAR THE ELAPSED TIME COUNTER  
MOV @2000., WATCH ;SET WATCH TO 2000 MS

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

;SEIZE THE DRIVE THROUGH PORT A

(2) 004352 113760 001216 000010  
(2) 004360 013737 001216 001230  
(2) 004366 005060 000012  
(3) 004372 113760 001220 000010  
(3) 004400 013737 001220 001226  
(2) 004406 013737 001220 001232  
(2) 004414 016037 000012 001126  
(2) 004422 010037 001122  
(2) 004426 062737 000012 001122  
(2) 004434 005037 001124  
(2) 004440 023737 001124 001126  
(2) 004446 001403  
(2) 004450 104004  
(2) 004452 000137 005634  
(3) 004456 113760 001216 000010  
(3) 004464 013737 001216 001226

MOVB PORTA, RHCS2(RO) ;SELECT PORT A  
MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS  
CLR RHDS1(RO) ;WRITE RHDS1  
MOVB PORTB, RHCS2(RO) ;SELECT PORT B  
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS  
MOV RHDS1(RO), \$BDDAT ;SEE IF DRIVE SEIZED BY PORT A  
MOV RO, \$BADDR ;RH11 BASE ADDRESS  
ADD @RHDS1, \$BADDR ;GENERATE BAD REGISTER ADDRESS  
CLR \$GDDAT ;REGISTER SHOULD BE ZERO  
CMP \$GDDAT, \$BDDAT ;IS THE REGISTER ZERO  
BEQ .+10 ;BR IF IT IS  
ERROR 4 ;REPORT THE ERROR  
JMP \$S ;BYPASS REST OF THE SUBTEST  
MOVB PORTA, RHCS2(RO) ;SELECT PORT A  
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

(2) 004472 016037 000012 001126    MOV    RHDS1(RO),SBDDAT    ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 004500 012737 011600 001124    MOV    #MOL!PGM!DPR!DRY,SGDDAT    ;EXPECTED STATUS
(2) 004506 013737 001124 001160    MOV    SGDDAT,STMP1    ;USE (000) DATA AS A MASK
(2) 004514 005137 001160    COM    STMP1    ;COMPLEMENT THE EXPECTED STATUS
(2) 004520 013737 001126 001156    MOV    SBDDAT,STMP0    ;SAVE THE ACTUAL STATUS
(2) 004526 043737 001160 001156    BIC    STMP1,STMP0    ;CLEAR UNWANTED BITS
(2) 004534 023737 001124 001156    CMP    SGDDAT,STMP0    ;ARE THE EXPECTED STATUS BITS SET
(2) 004544 104005    SET    ERROR 5    ;IF THEY ARE
                          ;REPORT THE ERROR

```

(1)  
(2) ;\*\*\*\*\*  
(1) ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK  
(1)

```

(2) 004546 113750 001220 000010    MOV    PORTB,RHCS2(RO)    ;SELECT PORT B
(2) 004554 013737 001220 001226    MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 004562 016046 000046    MOV    RHEC2(RO),-(SP)    ;STORE REGISTER RHEC2, PORT B, FOR CHECK
(2) 004566 016046 000044    MOV    RHEC1(RO),-(SP)    ;STORE REGISTER RHEC1, PORT B, FOR CHECK
(2) 004572 016046 000042    MOV    RHER3(RO),-(SP)    ;STORE REGISTER RHER3, PORT B, FOR CHECK
(2) 004576 016046 000030    MOV    RHSN(RO),-(SP)    ;STORE REGISTER RHSN, PORT B, FOR CHECK
(2) 004602 016046 000036    MOV    RHCC(RO),-(SP)    ;STORE REGISTER RHCC, PORT B, FOR CHECK
(2) 004606 016046 000034    MOV    RHCA(RO),-(SP)    ;STORE REGISTER RHCA, PORT B, FOR CHECK
(2) 004612 016046 000032    MOV    RHOF(RO),-(SP)    ;STORE REGISTER RHOF, PORT B, FOR CHECK
(2) 004616 016046 000040    MOV    RHER2(RO),-(SP)    ;STORE REGISTER RHER2, PORT B, FOR CHECK
(2) 004622 016046 000020    MOV    RHLA(RO),-(SP)    ;STORE REGISTER RHLA, PORT B, FOR CHECK
(2) 004626 016046 000026    MOV    RHD1(RO),-(SP)    ;STORE REGISTER RHD1, PORT B, FOR CHECK
(2) 004632 016046 000006    MOV    RHDA(RO),-(SP)    ;STORE REGISTER RHDA, PORT B, FOR CHECK
(2) 004636 016046 000024    MOV    RHRM(RO),-(SP)    ;STORE REGISTER RHRM, PORT B, FOR CHECK
(2) 004642 016046 000014    MOV    RHER1(RO),-(SP)    ;STORE REGISTER RHER1, PORT B, FOR CHECK

```

(1)  
(2) ;\*\*\*\*\*  
(1) ;WAIT FOR PORT A TO TIMEOUT  
(1)

```

(1) 004646 005760 000012    15: TST    RHDS1(RO)    ;WAIT FOR THE DRIVE TO TIMEOUT
(1) 004652 001006    BNE    25    ;OR WHEN TIMEOUT OCCURS
(1) 004654 005737 001246    TST    WATCH    ;CHECK WATCH
(1) 004660 001372    BNE    15    ;OR AT ZERO
(1) 004662 104036    ERROR 36    ;NO TIMEOUT - IN 2 SECONDS
(1) 004664 000137 005254    JMP    45    ;BYPASS TIMEOUT 1st CHECK
(1) 004670 012737 000340 177776 25: MOV    #7*32,3APS    ;SET PRIORITY TO 7 TO STOP CLOCK
(1) 004676 013737 001244 001250    MOV    TIME,TIMEA    ;SAVE THE ELAPSED TIME FOR PORT A
(1) 004704 004537 057450    TSR    RS,TOLER    ;CALCULATE THE TOLERANCE
(1) 004710 001250    .WORD  TIMEA    ;TIMEOUT VALUE FOR PORT A
(1) 004712 012637 001252    MOV    (SP)+,TIMEAP    ;+25% TOLERANCE
(1) 004716 012637 001254    MOV    (SP)+,TIMEAM    ;-25% TOLERANCE

```

(1)  
(2) ;\*\*\*\*\*  
(1) ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS  
(1)

```

(1) 004722 023727 001244 000764    CMP    TIME,450    ;WAS MEASURED TIME AT LEAST 500 MS?
(1) 004730 103001    THIS 35    ;OR IF IT WAS
(1) 004732 104055    ERROR 55    ;REPORT TIMEOUT TOO SHORT

```

(1)  
(2) ;\*\*\*\*\*  
(1) ;VERIFY THAT THE DRIVE RETURNED TO NORMAL AFTER A TIMEOUT  
(1)



```

(2) 004734 012737 000240 177776 3S: MOV #<5*32.>,R0PS ;RESTORE PRIORITY TO 5
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) 004742 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 004746 012737 000012 001122 MOV #RHDS1,$H0ADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 004754 060037 001122 ADD R0,$B0ADR ;ADD THE I/O BASE ADDRESS
(2) 004760 012737 011600 001124 MOV #M0L!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
(2) 004766 113760 001216 000010 MOV B PORTA,RHCS2(R0) ;SELECT PORT A.
(2) 004774 016037 000012 001162 MOV RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 005002 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
(2) 005010 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 005016 113760 001220 000010 MOV B PORTB,RHCS2(R0) ;SELECT PORT B.
(2) 005024 016037 000012 001164 MOV RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 005032 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
(2) 005040 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 005046 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 005054 001006 BNE 64S ;BR IF NOT
(2) 005056 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 005062 001037 BNE 66S ;BR IF NOT
(2) 005064 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 005066 000137 005252 JMP 69S ;BYPASS THE REST OF THE CHECKS
(2) 005072 013737 001162 001126 64S: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 005100 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 005106 113760 001220 000010 MOV B PORTB,RHCS2(R0) ;SELECT PORT B.
(2) 005114 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 005120 001414 BEQ 65S ;BR IF ZERO
(2) 005122 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 005130 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 005136 113760 001216 000010 MOV B PORTA,RHCS2(R0) ;SELECT PORT A.
(2) 005144 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 005150 001004 BNE 66S ;BR IF NOT
(2) 005152 012737 177777 001242 65S: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 005160 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 005162 013737 001162 001126 66S: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 005170 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 005176 042737 100100 001162 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 005204 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 005212 001401 BEQ 67S ;BR IF OK FROM PORT A.
(2) 005214 104007 ERROR 7 ;REPORT ERROR
(2) 005216 013737 001164 001126 67S: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 005224 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 005232 042737 100100 001164 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 005240 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 005246 001401 BEQ 68S ;BR IF OK
(2) 005250 104007 ERROR 7 ;REPORT ERROR
(2) 005252 000240 68S: NOP

(1) ;*****
(2) ;CHECK THE REGISTERS STORED THROUGH PORT B. ALL REGISTERS SHOULD BE ZERO.
(1) ;THE REGISTERS ARE STORED ON THE STACK.
(1) 005254 013737 001220 001226 4S: MOV PORTB,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT

```

(2)	005262	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHER1
(2)	005266	062737	000014	ADD	#RHER1, \$BDADR	:ADDRESS OF RHER1 FOR TYPEOUT
(2)	005274	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHER1
(2)	005300	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005302	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005304	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHMR
(2)	005310	062737	000024	ADD	#RHMR, \$BDADR	:ADDRESS OF RHMR FOR TYPEOUT
(2)	005316	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHMR
(2)	005322	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005324	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005326	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHDA
(2)	005332	062737	000006	ADD	#RHDA, \$BDADR	:ADDRESS OF RHDA FOR TYPEOUT
(2)	005340	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHDA
(2)	005344	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005346	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005350	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHDT
(2)	005354	062737	000026	ADD	#RHDT, \$BDADR	:ADDRESS OF RHDT FOR TYPEOUT
(2)	005362	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHDT
(2)	005364	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005370	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005372	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHLA
(2)	005376	062737	000020	ADD	#RHLA, \$BDADR	:ADDRESS OF RHLA FOR TYPEOUT
(2)	005404	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHLA
(2)	005410	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005412	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005414	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHER2
(2)	005420	062737	000040	ADD	#RHER2, \$BDADR	:ADDRESS OF RHER2 FOR TYPEOUT
(2)	005426	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHER2
(2)	005432	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005434	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005436	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHOF
(2)	005442	062737	000032	ADD	#RHOF, \$BDADR	:ADDRESS OF RHOF FOR TYPEOUT
(2)	005450	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHOF
(2)	005454	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005456	104006		ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005460	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHCA
(2)	005464	062737	000034	ADD	#RHCA, \$BDADR	:ADDRESS OF RHCA FOR TYPEOUT
(2)	005472	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHCA
(2)	005476	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005500	104006		ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	005502	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHCC
(2)	005506	062737	000036	ADD	#RHCC, \$BDADR	:ADDRESS OF RHCC FOR TYPEOUT
(2)	005514	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHCC
(2)	005520	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005522	104006		ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	005524	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHSN
(2)	005530	062737	000030	ADD	#RHSN, \$BDADR	:ADDRESS OF RHSN FOR TYPEOUT
(2)	005536	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHSN
(2)	005542	001401		BEQ	.+4	:CONTENTS ZERO ?
(2)	005544	104006		ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	005546	010037	001122	MOV	RO, \$BDADR	:BASE ADDRESS FOR REGISTER RHER3
(2)	005552	062737	000042	ADD	#RHER3, \$BDADR	:ADDRESS OF RHER3 FOR TYPEOUT
(2)	005560	012637	001126	MOV	(SP)+, \$BDDAT	:CHECK THE STORED CONTENTS OF RHER3
(2)	005564	001401		BEQ	.+4	:CONTENTS ZERO ?

```

(2) 005566 104006          ERROR 6          ;REPORT THAT PORT B SEES NON-ZERO REGISTER
(2) 005570 010037 001122  MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RHEC1
(2) 005574 062737 000044 001122  ADD     #RHEC1,$BDADR ;ADDRESS OF RHEC1 FOR TYPEOUT
(2) 005602 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHEC1
(2) 005606 001401          BEQ     .+4          ;CONTENTS ZERO ?
(2) 005610 104006          ERROR 6          ;REPORT THAT PORT B SEES NON-ZERO REGISTER
(2) 005612 010037 001122  MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RHEC2
(2) 005616 062737 000046 001122  ADD     #RHEC2,$BDADR ;ADDRESS OF RHEC2 FOR TYPEOUT
(2) 005624 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHEC2
(2) 005630 001401          BEQ     .+4          ;CONTENTS ZERO ?
(2) 005632 104006          ERROR 6          ;REPORT THAT PORT B SEES NON-ZERO REGISTER
(2) 005634

```

SS:

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 005634 105737 001103  TSTB   $ERFLG      ;DID AN ERROR OCCUR ?
(4) 005640 001412          BEQ     TST3        ;BR IF NOT
(2) 005642 032737 001000 177570  BIT    #SW09,$SWR   ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 005650 001406          BEQ     TST3        ;BR IF NOT
(2) 005652 105037 001103  CLRB   $ERFLG      ;CLEAR THE ERROR FLAG
(2) 005656 005037 001170  CLR    $TIMES       ;CLEAR THE MAX ITERATION COUNT
(2) 005662 000177 173222  JMP    @SLPERR      ;GO TO THE LOOP ADDRESS

```

5128  
5129

\*\*\*\*\*  
\*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 RHDS1 THROUGH PORT 'B'; VERIFY THAT THE DRIVE
*HAS BEEN SEIZED.
*
*B. READ EACH DRIVE REGISTER, EXCEPT RHCS1, 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.

```

```

(3) 005666          TST3:
(3) 005666 000004          SCOPE
(3) 005670 005737 001266  TST    KYBCTL      ;INITIALIZE THE SCOPE HANDLER
(3) 005674 001406          BEQ     2$         ;PERFORMING ONLY SINGLE TESTS ?
(3) 005676 100002          BPL    1$         ;BR IF NOT
(3) 005700 000137 002410  JMP    EXEC       ;BR IF JUST ENTERED TEST
(3) 005704 012737 177777 001266 1$:  MOV    #-1,KYBCTL  ;RETURN & GET NEXT TEST NUMBER
(3) 005712 112737 000003 001102 2$:  MOVB   #3,$STNM   ;SET SINGLE TEST INDICATOR
(3) 005720 012737 005742 001106  MOV    #TEST3,$LPADR ;TEST NUMBER
(3) 005726 012737 005742 001110  MOV    #TEST3,$LPERR ;LOAD LOOP ON TEST ADDRESS
(1) 005734 012737 000012 001170  MOV    #10,$TIMES  ;LOAD LOOP ON ERROR ADDRESS
(3)

```

5130  
(3)

\*\*\*\*\*

```

(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) TEST3:
(1) 005742 012737 000240 177776 MOV #<5*32.>,a#PS ;SET PRIORITY TO 5 IN CASE LOOPING
(1) 005750 005037 001256 CLR TIMEB ;CLEAR TIMEOUT VALUE FOR PORT B
(1) 005754 005037 001260 CLR TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
(1) 005760 005037 001262 CLR TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE
(2)
(3) ;*****
(2) ;START THE TIMER
(2)
(2) 005764 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 005770 012737 003720 001246 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
(1)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2)
(2) 005776 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 006004 013737 001220 001230 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 006012 005060 000012 CLR RHDS1(RO) ;WRITE RHDS1
(3) 006016 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 006024 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006032 013737 001216 001232 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 006040 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 006046 010037 001122 MOV RO,SBADR ;RH11 BASE ADDRESS
(2) 006052 062737 000012 001122 ADD #RHDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
(2) 006060 005037 001124 CLR SGDDAT ;REGISTER SHOULD BE ZERO
(2) 006064 023737 001124 001126 CMP SGDDAT,SBDDAT ;IS THE REGISTER ZERO
(2) 006072 001403 BEQ .+10 ;BR IF IT IS
(2) 006074 104004 ERROR 4 ;REPORT THE ERROR
(2) 006076 000137 007260 JMP 5$ ;BYPASS REST OF THE SUBTEST
(3) 006102 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 006110 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006116 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 006124 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,SGDDAT ;EXPECTED STATUS
(2) 006132 013737 001124 001160 MOV SGDDAT,STMP1 ;USE GOOD DATA AS A MASK
(2) 006140 005137 001160 COM STMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 006144 013737 001126 001156 MOV SBDDAT,STMPO ;SAVE THE ACTUAL STATUS
(2) 006152 043737 001160 001156 BIC STMP1,STMPO ;CLEAR UNWANTED BITS
(2) 006160 023737 001124 001156 CMP SGDDAT,STMPO ;ARE THE EXPECTED STATUS BITS SET ?
(2) 006166 001401 BEQ .+4 ;BR IF THEY ARE
(2) 006170 104005 ERROR 5 ;REPORT THE ERROR
(1)
(2) ;*****
(1) ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
(1)
(2) 006172 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 006200 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006206 016046 000046 MOV RHEC2(RO),-(SP) ;STORE REGISTER RHEC2, PORT A, FOR CHECK
(2) 006212 016046 000044 MOV RHEC1(RO),-(SP) ;STORE REGISTER RHEC1, PORT A, FOR CHECK
(2) 006216 016046 000042 MOV RHER3(RO),-(SP) ;STORE REGISTER RHER3, PORT A, FOR CHECK
(2) 006222 016046 000030 MOV RHSN(RO),-(SP) ;STORE REGISTER RHSN, PORT A, FOR CHECK
(2) 006226 016046 000036 MOV RHCC(RO),-(SP) ;STORE REGISTER RHCC, PORT A, FOR CHECK
    
```

```

(2) 006232 016046 000034      MOV      RHCA(RO),-(SP) ;STORE REGISTER RHCA, PORT A, FOR CHECK
(2) 006236 016046 000032      MOV      RHOF(RO),-(SP) ;STORE REGISTER RHOF, PORT A, FOR CHECK
(2) 006242 016046 000040      MOV      RHER2(RO),-(SP) ;STORE REGISTER RHER2, PORT A, FOR CHECK
(2) 006246 016046 000020      MOV      RHLA(RO),-(SP) ;STORE REGISTER RHLA, PORT A, FOR CHECK
(2) 006252 016046 000026      MOV      RHDT(RO),-(SP) ;STORE REGISTER RHDT, PORT A, FOR CHECK
(2) 006256 016046 000006      MOV      RHDA(RO),-(SP) ;STORE REGISTER RHDA, PORT A, FOR CHECK
(2) 006262 016046 000024      MOV      RHMR(RO),-(SP) ;STORE REGISTER RHMR, PORT A, FOR CHECK
(2) 006266 016046 000014      MOV      RHER1(RO),-(SP) ;STORE REGISTER RHER1, PORT A, FOR CHECK
(1)
(2)
(1) ;*****
(1) ;WAIT FOR PORT B TO TIMEOUT
(1)
(1) 006272 005760 000012      1$: TST      RHDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
(1) 006276 001006          BNE      2$ ;BR WHEN TIMEOUT OCCURS
(1) 006300 005737 001246      TST      WATCH ;CHECK WATCH
(1) 006304 001372          BNE      1$ ;BR IF NOT ZERO
(1) 006306 104036          ERROR    36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 006310 000137 006700      JMP      4$ ;BYPASS TIMEOUT TIME CHECK
(1) 006314 012737 000340 177776 2$: MOV      #(7*32.),@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
(1) 006322 013737 001244 001256      MOV      TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
(1) 006330 004537 057450      JSR      R5,TOLER ;CALCULATE THE TOLERANCE
(1) 006334 001256          .WORD    TIMEB ;TIMEOUT VALUE FOR PORT B
(1) 006336 012637 001260      MOV      (SP)+,TIMEBP ;+25% TOLERANCE
(1) 006342 012637 001262      MOV      (SP)+,TIMEBM ;-25% TOLERANCE
(1)
(2) ;*****
(1) ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
(1)
(1) 006346 023727 001244 000764      CMP      TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
(1) 006354 103001          BHS      3$ ;BR IF IT WAS
(1) 006356 104055          ERROR    55 ;REPORT TIMEOUT TOO SHORT
(1)
(2) ;*****
(1) ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT
(1)
(1) 006360 012737 000240 177776 3$: MOV      #(5*32.),@#PS ;RESTORE PRIORITY TO 5
(2)
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 006366 005037 001242          CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 006372 012737 000012 001122      MOV      #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 006400 060037 001122          ADD      RO,$BDADR ;ADD THE I/O BASE ADDRESS
(2) 006404 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
(2) 006412 113760 001216 000010      MOVVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 006420 016037 000012 001162      MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 006426 013737 001162 001156      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
(2) 006434 042737 100100 001156      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 006442 113760 001220 000010      MOVVB   PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 006450 016037 000012 001164      MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 006456 013737 001164 001160      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
(2) 006464 042737 100100 001160      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 006472 023737 001156 001160      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 006500 001006          BNE      64$ ;BR IF NOT
(2) 006502 005737 001156          TST      STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
    
```

```

(2) 006506 001037 BNE 66$ ;BR IF NOT
(2) 006510 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 006512 000137 006676 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(2) 006516 013737 001162 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 006524 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 006532 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 006540 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 006544 001414 BEQ 65$ ;BR IF ZERO
(2) 006546 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 006554 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 006562 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 006570 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 006574 001004 BNE 66$ ;BR IF NOT
(2) 006576 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 006604 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 006606 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 006614 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 006622 042737 100100 001162 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 006630 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 006636 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(2) 006640 104007 ERROR 7 ;REPORT ERROR
(2) 006642 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 006650 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 006656 042737 100100 001164 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 006664 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 006672 001401 BEQ 68$ ;BR IF OK
(2) 006674 104007 ERROR 7 ;REPORT ERROR
(2) 006676 000240 68$: NOP

```

```

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

```

```

(1) 006700 013737 001216 001226 4$: MOV PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
(2) 006706 010037 001122 MOV RO,$BDDADR ;BASE ADDRESS FOR REGISTER RHER1
(2) 006712 062737 000014 001122 ADD #RHER1,$BDDADR ;ADDRESS OF RHER1 FOR TYPEOUT
(2) 006720 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHER1
(2) 006724 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 006726 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 006730 010037 001122 MOV RO,$BDDADR ;BASE ADDRESS FOR REGISTER RHMR
(2) 006734 062737 000024 001122 ADD #RHMR,$BDDADR ;ADDRESS OF RHMR FOR TYPEOUT
(2) 006742 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHMR
(2) 006746 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 006750 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 006752 010037 001122 MOV RO,$BDDADR ;BASE ADDRESS FOR REGISTER RHDA
(2) 006756 062737 000006 001122 ADD #RHDA,$BDDADR ;ADDRESS OF RHDA FOR TYPEOUT
(2) 006764 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHDA
(2) 006770 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 006772 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 006774 010037 001122 MOV RO,$BDDADR ;BASE ADDRESS FOR REGISTER RHDT
(2) 007000 062737 000026 001122 ADD #RHDT,$BDDADR ;ADDRESS OF RHDT FOR TYPEOUT
(2) 007006 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RHDT
(2) 007012 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007014 104006 ERROR 6 ;REPORT THAT PORT A SAW NON-ZERO REGISTER

```



```

(2) 007302 005037 001170 CLR STIMES ;CLEAR THE MAX ITERATION COUNT
(2) 007306 000177 171576 JMP $SLPERR ;GO TO THE LOOP ADDRESS
    
```

S131  
S149  
S150

```

(3) *****
(3) *TEST 4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
(4) *
(4) *VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
(4) * FOR THE PORT UNDER TEST.
(4) *
(4) * A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE
(4) * DRIVE WAS SEIZED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
(4) *
(4) * B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE
(4) * 'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
(4) * FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
(4) * 'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
(4) *
(4) * C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
(4) * THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
(4) *
(3) *****
    
```

```

(3) 007312
(3) 007312 000004
(3) 007314 005737 001266 SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 007320 001406 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 007322 100002 BEQ 25 ;BR IF NOT
(3) 007324 000137 002410 BPL 15 ;BR IF JUST ENTERED TEST
(3) 007330 012737 177777 001266 15: JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 007336 112737 000004 001102 25: MOV #1, KYBCTL ;SET SINGLE TEST INDICATOR
(3) 007344 012737 007366 001106 MOV #4, $STNM ;TEST NUMBER
(3) 007352 012737 007366 001110 MOV #TEST4, $LPADR ;LOAD LOOP ON TEST ADDRESS
(1) 007360 012737 000001 001170 MOV #TEST4, $LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) MOV #1, $TIMES ;DO 1 ITERATION
    
```

S151  
S202

```

(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2) *****
    
```

```

(2) 007366
(2) 007366 113760 001216 000010 TEST4: MOV#B PORTA, RHCS2(RO) ;SELECT PORT A
(2) 007374 013737 001216 001226 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) *****
    
```

```

(3) *****
(2) ;START THE TIMER
(2) *****
    
```

```

(2) 007402 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 007406 012737 003720 001246 MOV #2000, WATCH ;SET WATCH TO 2000 MS
(1) 007414 013737 001216 001230 MOV PORTA, SEIZPT ;'SEIZED' PORT ADDRESS
(1) *****
    
```

```

(2) *****
(1) ;ISSUE DRIVE CLEAR COMMAND
(1) *****
    
```

```

(1) 007422 012760 000011 000000 MOV #11, RHCS1(RO) ;ISSUE A DRIVE CLEAR
(1)
    
```



```

(2) ;*****
(1) ;VERIFY THAT DRIVE SEIZED BY PORT A.
(1)
(2) 007430 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 007436 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 007444 005037 001236                CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007450 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 007456 012737 000012 001122      MOV    #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 007464 060037 001122                ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 007470 005037 001124                CLR    $GDDAT ;WHAT REGISTER SHOULD BE
(2) 007474 023737 001124 001126      CMP    $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 007502 001403                BEQ    645 ;BR IF OK
(2) 007504 104012                ERROR  12 ;TYPE MESSAGE 12
(2) 007506 005137 001236                COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 007512 000240                NOP
(2) 007514 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 007522 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 007530 005037 001236                CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007534 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 007542 012737 000012 001122      MOV    #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 007550 060037 001122                ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 007554 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 007562 013737 001126 001156      MOV    SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 007570 042737 106177 001156      BIC    #1C71600,STMP0 ;SAVE SPECIFIED BITS
(2) 007576 023737 001124 001156      CMP    $GDDAT,STMP0 ;COMPARE THE BITS
(2) 007604 001414                BEQ    655 ;BR IF OK
(2) 007606 013737 001126 001166      MOV    $BDDAT,STMP4 ;COPY 'BAD DATA'
(2) 007614 042737 071600 001166      BIC    #71600,STMP4 ;CLEAR THE MASKED BITS
(2) 007622 053737 001166 001124      BIS    STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 007630 104010                ERROR  10 ;REPORT THE ERROR
(2) 007632 005137 001236                COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 007636 000240                NOP
(2) 007640 005037 001236                CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007644 016037 000000 001126      MOV    RHCS1(RO),SBDDAT ;GET CONTENTS OF RHCS1
(2) 007652 012737 000000 001122      MOV    #RHCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 007660 060037 001122                ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 007664 012737 004210 001124      MOV    #4210,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 007672 013737 001126 001156      MOV    SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 007700 042737 100000 001156      BIC    #1C77777,STMP0 ;SAVE SPECIFIED BITS
(2) 007706 023737 001124 001156      CMP    $GDDAT,STMP0 ;COMPARE THE BITS
(2) 007714 001414                BEQ    665 ;BR IF OK
(2) 007716 013737 001126 001166      MOV    $BDDAT,STMP4 ;COPY 'BAD DATA'
(2) 007724 042737 077777 001166      BIC    #77777,STMP4 ;CLEAR THE MASKED BITS
(2) 007732 053737 001166 001124      BIS    STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 007740 104010                ERROR  10 ;REPORT THE ERROR
(2) 007742 005137 001236                COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 007746 000240                NOP
(1)
(2) ;*****
(1) ;ISSUE READIN PRESET COMMAND AND SET FMT22
(1)
(1) 007750 012760 000023 000000      MOV    #23,RHCS1(RO) ;ISSUE A READIN PRESET
(1) 007756 012760 010000 000032      MOV    #FMT22,RHOF(RO) ;SET FMT22

```

F05

(2) ;\*\*\*\*\*  
(1) ;VERIFY THAT THE DRIVE STATUS IS CORRECT

```

(2) 007764 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007770 016037 000012 001125 MOV RHDS1(R0), $BDDAT ;GET CONTENTS OF RHDS1
(2) 007776 012737 000012 001122 MOV #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 010004 005037 001122 ADD R0, $BDADR ;ADD RHL1 BASE ADDRESS
(2) 010010 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 010016 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 010024 042737 100177 001156 BIC #1C71700, $TMP0 ;SET SPECIFIED BITS
(2) 010032 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 010040 001414 001126 001156 BEQ 675 ;BR IF OK
(2) 010042 013737 001126 001156 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 010050 042737 071700 001156 BIC #71700, $TMP4 ;CLEAR THE MASKED BITS
(2) 010056 053737 001166 001124 BIS $TMP4, $GDDAT ;SET WITH GOOD DATA FOR TYPEOUT
(2) 010054 104013 001166 001124 ERROR 13 ;MESSAGE 13
(2) 010066 005137 001236 CLR CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 010072 000240 675: NOP ;
(2) 010074 113760 001220 000010 MOVE PORTA, RHCS2(R0) ;SELECT PORT B
(2) 010102 013737 001220 001226 MOV PORTB, $TMPR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

(1) ;\*\*\*\*\*  
(2) ;WAIT FOR TIMEOUT TO RELEASE DRIVE

```

(1) 010110 005760 000012 15: TST RHDS1(R0) ;WAIT FOR THE PORT TO TIME OUT
(1) 010114 001006 BNE 25 ;OR WHEN TIMEOUT OCCURS
(1) 010116 005737 001246 TST WATCH ;CHECK THE WATCH
(1) 010122 001372 BNE 15 ;CHECK NOT ZERO
(1) 010124 104036 ERROR 36 ;BR IF NOT HIT IN 2 SECONDS
(1) 010126 000137 010444 JMP 35 ;NO TIMEOUT - REGISTER CHECK

```

(1) ;\*\*\*\*\*  
(2) ;SEE IF DRIVE RETURNED TO NEUTRAL

```

(2) 010132 25: ;
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL.
(2) 010132 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 010136 012737 000012 001122 MOV #RHDS1, $BDADR ;FORM THE ADDRESS OF RHDS1
(2) 010144 060037 001122 ADD R0, $BDADR ;ADD THE I/O BASE ADDRESS
(2) 010150 012737 011600 001124 MOV #MOL!PCM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
(2) 010156 113760 001216 000010 MOVB PORTA, RHCS2(R0) ;SELECT PORT A.
(2) 010164 016037 000012 001162 MOV RHDS1(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 010172 013737 001162 001156 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
(2) 010200 042737 100100 001156 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 010206 113760 001220 000010 MOVB PORTB, RHCS2(R0) ;SELECT PORT B.
(2) 010214 016037 000012 001164 MOV RHDS1(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 010222 013737 001164 001160 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
(2) 010230 042737 100100 001160 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 010236 023737 001156 001160 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 010244 001006 BNE 685 ;BR IF NOT
(2) 010246 005737 001156 TST $TMP0 ;ARE THEY ZERO ?
(2) 010252 001037 BNE 705 ;BR IF NOT

```





```

(2) 010740 012737 011600 00117
(2) 010746 013737 001177 00117
(2) 010754 042737 106177 00117
(2) 010762 023737 001174 00117
(2) 010770 001414
(2) 010772 013737 001126
(2) 011000 042737 071600
(2) 011005 053737 001166
(2) 011014 104010
(2) 011016 005137 00123
(2) 011022 000240
(2) 011024 005037 00123
(2) 011030 016037 001126
(2) 011036 012737 001122
(2) 011044 060037 001122
(2) 011050 012737 004211 11124
(2) 011056 013737 001126
(2) 011064 042737 100027 11124
(2) 011072 053737 001126
(2) 011100 001414
(2) 011102 013737 001126 001126
(2) 011110 042737 077777 001126
(2) 011116 053737 001166 001126
(2) 011124 104010
(2) 011126 005137 001236
(2) 011132 000240
(1)
(1)
(1) 011134 011760 000023 0000
(1) 011142 012760 010000 000032
(2)
(2)
(2)
(2) 011170 016037 000012 001126
(2) 011174 012737 000012 001122
(2) 011174 060037 001122
(2) 011174 012737 011700 001124
(2) 011202 013737 001126 001156
(2) 011210 042737 106077 001156
(2) 011216 023737 001124 001156
(2) 011224 001414
(2) 011234 012737 001126 001156
(2) 011234 042737 071700 001160
(2) 011242 053737 001166 001124
(2) 011250 104013
(2) 011252 005137 001236
(2) 011256 000240
(2) 011260 113760 001216 000010
(2) 011266 013737 001216 001226
(1)

```

```

MOV #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1C71600,$STMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$STMP0 ;COMPARE THE BITS
BEQ 655 ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #71600,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHCS1(RO),$BDDAT ;GET CONTENTS OF RHCS1
MOV #RHCS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
MOV #4210,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #77777,$STMP0 ;SAVE SPECIFIED BITS
CMP $BDDAT,$STMP0 ;COMPARE THE BITS
BEQ 655 ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #77777,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
MOVB FORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

655:

\*\*\*\*\*  
:ISSUE READIN PRESET COMMAND AND SET FMT22

```

MOV #23,RHCS1(RO) ;ISSUE A READIN PRESET
MOV #FMT22,RHOF(RO) ;SET FMT22

```

\*\*\*\*\*  
:VERIFY THAT THE DRIVE STATUS IS CORRECT

```

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1C71700,$STMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$STMP0 ;COMPARE THE BITS
BEQ 675 ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #71700,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 13 ;TYPE MESSAGE 13
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
MOVB FORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

675:

```

(2) ;*****
(1) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1)
(1) 011274 005760 000012 1$: TST RHDS1(RO) ;WAIT FOR THE PORT TO TIME OUT
(1) 011300 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
(1) 011302 005737 001246 TST WATCH ;CHECK THE WATCH
(1) 011306 001372 BNE 1$ ;BR IF NOT ZERO
(1) 011310 104036 ERROR 3$ ;NO TIMEOUT WITHIN 2 SECONDS
(1) 011312 000137 011630 JMP 3$ ;BYPASS ATTN REGISTER CHECK
(1)
(2) ;*****
(1) ;SEE IF DRIVE RETURNED TO NEUTRAL
(1)
(2) 011316 2$:
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 011316 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 011322 012737 000012 001122 MOV #RHDS1,$BDDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 011330 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(2) 011334 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 011342 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 011350 016037 000012 001162 MOV RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 011356 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 011364 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 011372 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 011400 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 011406 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 011414 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 011422 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 011430 001006 BNE 68$ ;BR IF NOT
(2) 011432 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 011436 001037 BNE 70$ ;BR IF NOT
(2) 011440 104046 ERROR 4$ ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 011442 000137 011626 JMP 72$ ;BYPASS THE REST OF THE CHECKS
(2) 011446 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 011454 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 011462 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 011470 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 011474 001414 BEQ 69$ ;BR IF ZERO
(2) 011476 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 011504 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 011512 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 011520 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 011524 001004 BNE 70$ ;BR IF NOT
(2) 011526 012737 177777 001242 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 011534 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(2) 011536 013737 001162 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 011544 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 011552 042737 100000 001162 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(2) 011560 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 011566 001401 BEQ 71$ ;BR IF OK FROM PORT A.
(2) 011570 104007 ERROR 7 ;REPORT ERROR
(2) 011572 013737 001164 001126 71$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.

```

```

(2) 011600 013737 001220 001226      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
(2) 011606 042737 100000 001164      BIC      #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
(2) 011614 023737 001124 001164      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(2) 011622 001401                      BEQ      72$              ;BR IF OK
(2) 011624 104007                      ERROR    7                ;REPORT ERROR
(2) 011626 000240                      72$:   NOP
(2) 011630                               3$:
(2)                                     ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2) 011630 105737 001103      TSTB     $ERFLG          ;DID AN ERROR OCCUR ?
(4) 011634 001412                      BEQ      TST6            ;:BR IF NOT
(2) 011636 032737 001000 177570      BIT      #SW09,$SWR      ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 011644 001406                      BEQ      TST6            ;:BR IF NOT
(2) 011646 105037 001103      CLRB     $ERFLG          ;CLEAR THE ERROR FLAG
(2) 011652 005037 001170      CLR      $TIMES          ;CLEAR THE MAX ITERATION COUNT
(2) 011656 000177 167226      JMP      @SLPERR         ;GO TO THE LOOP ADDRESS

```

5227  
5239  
5240  
(3)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(3)

```

*****
*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 RHDS1.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
*    RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
*    DRIVE.
*****

```

(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(1)  
5241  
5272

```

(3) 011662                TST6:
(3) 011662 000004          SCOPE          ;INITIALIZE THE SCOPE HANDLER
(3) 011664 005737 001266  TST      KYBCTL     ;PERFORMING ONLY SINGLE TESTS ?
(3) 011670 001406          BEQ      2$        ;BR IF NOT
(3) 011672 100002          BPL      1$        ;BR IF JUST ENTERED TEST
(3) 011674 000137 002410  JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
(3) 011700 012737 177777 001266 1$:   MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 011706 112737 000006 001102 2$:   MOV      #6,$STNM  ;TEST NUMBER
(3) 011714 012737 011736 001106   MOV      #TEST6,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 011722 012737 011736 001110   MOV      #TEST6,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 011730 012737 007640 001170   MOV      #4000.,$TIMES ;:DO 4000. ITERATIONS

```

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

(3)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)

```

(2) 011736                TEST6:
(2)
(2)
(3)
(2)
(2)
(2) 011736 005037 001244      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 011742 012737 003720 001246   MOV      #2000.,WATCH ;SET WATCH TO 2000 MS

```

```

(1)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT A
(2) 011750 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 011756 013737 001216 001230      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 011764 005060 000012              CLR    RHDS1(RO) ;WRITE RHDS1
(2) 011770 013737 001220 001232      MOV    PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(1)
(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT A
(3) 011776 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(3) 012004 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 012012 012760 000013 000000      MOV    #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 012020 005037 001242              CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 012024 012737 000012 001122      MOV    #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 012032 060037 001122              ADD    RO,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 012036 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 012044 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 012052 016037 000012 001162      MOV    RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 012060 013737 001162 001156      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 012066 042737 100100 001156      BIC    #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 012074 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 012102 016037 000012 001164      MOV    RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 012110 013737 001164 001160      MOV    STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 012116 042737 100100 001160      BIC    #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 012124 023737 001156 001160      CMP    STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 012132 001006                      BNE    64$ ;BR IF NOT
(3) 012134 005737 001156                      TST    STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 012140 001037                      BNE    66$ ;BR IF NOT
(3) 012142 104046                      ERROR  46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 012144 000137 012330                      JMP    68$ ;BYPASS THE REST OF THE CHECKS
(3) 012150 013737 001162 001126 64$: MOV    STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 012156 013737 001220 001226      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 012164 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 012172 005737 001156                      TST    STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 012176 001414                      BEQ    65$ ;BR IF ZERO
(3) 012200 013737 001216 001226      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 012206 013737 001164 001126      MOV    STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 012214 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 012222 005737 001160                      TST    STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 012226 001004                      BNE    66$ ;BR IF NOT
(3) 012230 012737 177777 001242 65$: MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 012236 104026                      ERROR  26 ;TYPE ERROR MESSAGE 26
(3) 012240 013737 001162 001126 66$: MOV    STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 012246 013737 001216 001226      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 012254 042737 100000 001162      BIC    #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
(3) 012262 023737 001124 001162      CMP    $GDDAT,STMP2 ;ALL BITS OK ?

```



```

(3) 012270 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(3) 012272 104007 ERROR 7 ;REPORT ERROR
(3) 012274 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 012302 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 012310 042737 100000 001164 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 012316 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 012324 001401 BEQ 68$ ;BR IF OK
(3) 012326 104007 ERROR 7 ;REPORT ERROR
(3) 012330 000240 68$: NOP
(1) 012332 005737 001242 TST RELERR ;DID DRIVE RETURN TO NEUTRAL ?
(1) 012336 001402 BEQ .+6 ;BR IF IN NEUTRAL
(1) 012340 000137 012614 JMP 1$ ;GO WAIT FOR DRIVE TO TIMEOUT
(2) 012344 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 012352 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 012360 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 012364 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 012372 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 012400 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 012404 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 012410 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 012416 042737 077777 001156 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 012424 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 012432 001414 BEQ 69$ ;BR IF OK
(2) 012434 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 012442 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 012450 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 012456 104017 ERROR 17 ;TYPE MESSAGE 17
(2) 012460 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 012464 000240 69$: NOP
(2) 012466 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 012474 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 012502 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 012506 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 012514 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 012522 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 012526 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 012532 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 012540 042737 077777 001156 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 012546 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 012554 001414 BEQ 70$ ;BR IF OK
(2) 012556 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 012564 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 012572 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 012600 104017 ERROR 17 ;TYPE MESSAGE 17
(2) 012602 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 012606 000240 70$: NOP
(1) 012610 000137 012646 JMP 2$ ;GO CHECK FOR LOOP ON ERROR
(1)
(2) ;*****
(1) ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
(1) ;TO RELEASE THE DRIVE
(1)
(2) 012614 1$:
(2) 012614 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B

```

```

(2) 012622 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 012630 005760 000012              TST    RHDS1(RO)   ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1) 012634 001004                      BNE    2$         ;BR WHEN DRIVE RELEASED
(1) 012636 005737 001246              TST    WATCH      ;CHECK THE WATCH
(1) 012642 001364                      BNE    1$         ;BR IF NOT ZERO
(1) 012644 104036                      ERROR   36        ;NO TIMEOUT WITHIN 2 SECONDS
(2) 012646                          2$:
(2)
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 012646 105737 001103              TSTB   SERFLG     ;DID AN ERROR OCCUR ?
(4) 012652 001412                      BEQ    TST7       ;BR IF NOT
(2) 012654 032737 001000 177570      BIT    #SW09,SWR  ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 012662 001406                      BEQ    TST7       ;BR IF NOT
(2) 012664 105037 001103              CLRB   SERFLG     ;CLEAR THE ERROR FLAG
(2) 012670 005037 001170              CLR    $TIMES     ;CLEAR THE MAX ITERATION COUNT
(2) 012674 000177 166210              JMP    @SLPERR    ;GO TO THE LOOP ADDRESS

```

5284  
5285

```

(3) *****
(3) *TEST 7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'
(4) *
(4) *TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
(4) *
(4) * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
(4) *
(4) * B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
(4) * RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
(4) * DRIVE.
(4) *
(3) *****

```

```

(3) 012700 TST7:
(3) 012700 000004      SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 012702 005737 001266 TST    KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 012706 001406      BEQ    2$     ;BR IF NOT
(3) 012710 100002      BPL    1$     ;BR IF JUST ENTERED TEST
(3) 012712 000137 002410 JMP    EXEC   ;RETURN & GET NEXT TEST NUMBER
(3) 012716 012737 177777 001266 1$: MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 012724 112737 000007 001102 2$: MOVB  #7,$TSTNM ;TEST NUMBER
(3) 012732 012737 012754 001106 MOV    #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 012740 012737 012754 001110 MOV    #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 012746 012737 007640 001170 MOV    #4000,$TIMES ;DO 4000. ITERATIONS

```

5286  
5287

```

(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)

```

```

(2) 012754 TEST7:
(2)
(3) *****
(2) ;START THE TIMER
(2)
(2) 012754 005037 001244      CLR    TIME      ;CLEAR THE ELAPSED TIME COUNTER
(2) 012760 012737 003720 001246 MOV    #2000.,WATCH ;SET WATCH TO 2000 MS
(1)

```

```

(3) ;*****
(3) ;SEIZE THE DRIVE THROUGH PORT B
(3) 012766 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 012774 013737 001220 001230 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(3) 013002 005060 000012 CLR RHDS1(RO) ;WRITE RHDS1
(3) 013006 013737 001216 001232 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(3) ;*****
(3) ;RELEASE THE DRIVE FROM PORT B
(3) 013014 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 013022 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 013030 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 013036 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 013042 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 013050 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 013054 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 013062 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 013070 016037 000012 001162 MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 013076 013737 001162 001156 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 013104 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 013112 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 013120 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 013126 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 013134 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 013142 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 013150 001006 BNE 64$ ;BR IF NOT
(3) 013152 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 013156 001037 BNE 66$ ;BR IF NOT
(3) 013160 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 013162 000137 013346 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(3) 013166 013737 001162 001126 64$: MOV STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 013174 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 013202 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 013210 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 013214 001414 BEQ 65$ ;BR IF ZERO
(3) 013216 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 013224 013737 001164 001126 MOV STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 013232 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT FORT A.
(3) 013240 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 013244 001004 BNE 66$ ;BR IF NOT
(3) 013246 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 013254 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 013256 013737 001162 001126 66$: MOV STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 013264 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 013272 042737 100000 001162 BIC #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
(3) 013300 023737 001124 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 013306 001401 BEQ 67$ ;BR IF OK FROM PORT A.

```

```

(3) 013310 104007          ERROR 7          ;REPORT ERROR
(3) 013312 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 013320 013737 001220 001226      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 013326 042737 100000 001164      BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 013334 023737 001124 001164      CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 013342 001401          BEQ 68$          ;BR IF OK
(3) 013344 104007          ERROR 7          ;REPORT ERROR
(3) 013346 000240          NOP              ;
(1) 013350 005737 001242          TST RELERR      ;DID DRIVE RETURN TO NEUTRAL ?
(1) 013354 001402          BEQ .+6         ;BR IF IN NEUTRAL
(1) 013356 000137 013632          JMP 1$         ;GO WAIT FOR DRIVE TO TIMEOUT
(2) 013362 113760 001220 000010      MOVVB PORTA,RHCS2(RO) ;SELECT PORT B
(2) 013370 013737 001220 001226      MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 013376 005037 001236          CLR CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 013402 016037 000012 001126      MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 013410 012737 000012 001122      MOV #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 013416 060037 001122          ADD RO,$BDADR  ;ADD RH11 BASE ADDRESS
(2) 013422 005037 001124          CLR $GDDAT     ;WHAT REGISTER SHOULD BE
(2) 013426 013737 001126 001156      MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 013434 042737 077777 001156      BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 013442 023737 001124 001156      CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 013450 001414          BEQ 69$          ;BR IF OK
(2) 013452 013737 001126 001166      MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 013460 042737 100000 001166      BIC #ATA,$TMP4  ;CLEAR THE MASKED BITS
(2) 013466 053737 001166 001124      BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 013474 104017          ERROR 17       ;TYPE MESSAGE 17
(2) 013476 005137 001236          COM CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 013502 000240          NOP              ;
(2) 013504 113760 001216 000010      MOVVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 013512 013737 001216 001226      MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 013520 005037 001236          CLR CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 013524 016037 000012 001126      MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 013532 012737 000012 001122      MOV #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 013540 060037 001122          ADD RO,$BDADR  ;ADD RH11 BASE ADDRESS
(2) 013544 005037 001124          CLR $GDDAT     ;WHAT REGISTER SHOULD BE
(2) 013550 013737 001126 001156      MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 013556 042737 077777 001156      BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 013564 023737 001124 001156      CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 013572 001414          BEQ 70$          ;BR IF OK
(2) 013574 013737 001126 001166      MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 013602 042737 100000 001166      BIC #ATA,$TMP4  ;CLEAR THE MASKED BITS
(2) 013610 053737 001166 001124      BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 013616 104017          ERROR 17       ;TYPE MESSAGE 17
(2) 013620 005137 001236          COM CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 013624 000240          NOP              ;
(1) 013626 000137 013664          JMP 2$         ;GO CHECK FOR LOOP ON ERROR
(1)
(2)
(1)
(1)
(1)
(2) 013632          1$:
(2) 013632 113760 001216 000010      MOVVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 013640 013737 001216 001226      MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

(1) 013646 005760 000012 TST RHDS1(RO) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1) 013652 001004 BNE 2$ ;BR WHEN DRIVE RELEASED
(1) 013654 005737 001246 TST WATCH ;CHECK THE WATCH
(1) 013660 001364 BNE 1$ ;BR IF NOT ZERO
(1) 013662 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(2) 013664 2$:
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2) 013664 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 013670 001412 BEQ TST10 ;:BR IF NOT
(2) 013672 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 013700 001406 BEQ TST10 ;:BR IF NOT
(2) 013702 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 013706 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 013712 000177 165172 JMP @SLPERR ;GO TO THE LOOP ADDRESS

```

5288  
5297  
5298

```

(3) *****
(3) *TEST 10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
(4) *
(4) *TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
(4) *
(4) * A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
(4) * NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
(4) *
(3) *****

```

```

(3) TST10:
(3) 013716 000004 SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 013720 005737 001266 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 013724 001406 BEQ 2$ ;BR IF NOT
(3) 013726 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 013730 000137 002410 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 013734 012737 177777 001266 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 013742 112737 000010 001102 2$: MOVB #10,$TSTNM ;TEST NUMBER
(3) 013750 012737 013772 001106 MOV #TEST10,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 013756 012737 013772 001110 MOV #TEST10,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 013764 012737 000144 001170 MOV #100.,$TIMES ;DO 100. ITERATIONS

```

5299  
5316

```

(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2) TEST10:
(2) 013772 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 014000 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 014006 013737 001216 001230 MOV PORTA,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
(1) *****
(1) ;ISSUE A RELEASE COMMAND
(1) 014014 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE A RELEASE COMMAND
(1) *****
(2) ;

```

```

(1)                                     ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
(1)
(2)
(2)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) 014022 005037 001242                CLR      RELERR                ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 014026 012737 000012 001122        MOV      #RHDS1,$BDDADR        ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 014034 060037 001122                ADD      R0,$BDDADR            ;ADD THE I/O BASE ADDRESS
(2) 014040 012737 011700 001124        MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 014046 113760 001216 000010        MOVVB   PORTA,RHCS2(R0)        ;SELECT PORT A.
(2) 014054 016037 000012 001162        MOV      RHDS1(R0),$TMP2       ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 014062 013737 001162 001156        MOV      $TMP2,$TMP0          ;COPY IT INTO '$TMP0'
(2) 014070 042737 100100 001156        BIC      #ATA!VV,$TMP0        ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 014076 113760 001220 000010        MOVVB   PORTB,RHCS2(R0)        ;SELECT PORT B.
(2) 014104 016037 000012 001164        MOV      RHDS1(R0),$TMP3       ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 014112 013737 001164 001160        MOV      $TMP3,$TMP1          ;COPY IT INTO '$TMP1'
(2) 014120 042737 100100 001160        BIC      #ATA!VV,$TMP1        ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 014126 023737 001156 001160        CMP      $TMP0,$TMP1          ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 014134 001006                       BNE      64$                  ;BR IF NOT
(2) 014136 005737 001156                TST      $TMP0                ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 014142 001037                       BNE      66$                  ;BR IF NOT
(2) 014144 104046                       ERROR    46                   ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 014146 000137 014332                JMP      68$                  ;BYPASS THE REST OF THE CHECKS
(2) 014152 013737 001162 001126 64$:   MOV      $TMP2,$BDDAT          ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 014160 013737 001220 001226        MOV      PORTB,PTNBR          ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014166 113760 001220 000010        MOVVB   PORTB,RHCS2(R0)        ;SELECT PORT B.
(2) 014174 005737 001156                TST      $TMP0                ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 014200 001414                       BEQ      65$                  ;BR IF ZERO
(2) 014202 013737 001216 001226        MOV      PORTA,PTNBR          ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014210 013737 001164 001126        MOV      $TMP3,$BDDAT          ;'BAD DATA' FOR ERROR TYPE OUT
(2) 014216 113760 001216 000010        MOVVB   PORTA,RHCS2(R0)        ;SELECT PORT A.
(2) 014224 005737 001160                TST      $TMP1                ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 014230 001004                       BNE      66$                  ;BR IF NOT
(2) 014232 012737 177777 001242 65$:   MOV      #-1,RELERR           ;SET 'RELEASE ERROR' INDICATOR
(2) 014240 104030                       ERROR    30                   ;TYPE ERROR MESSAGE 30
(2) 014242 013737 001162 001126 66$:   MOV      $TMP2,$BDDAT          ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 014250 013737 001216 001226        MOV      PORTA,PTNBR          ;CHANGE PORT NUMBER
(2) 014256 042737 100000 001162        BIC      #ATA,$TMP2           ;DON'T CHECK THE ATTN BIT
(2) 014264 023737 001124 001162        CMP      $GDDAT,$TMP2         ;ALL BITS OK ?
(2) 014272 001401                       BEQ      67$                  ;BR IF OK FROM PORT A.
(2) 014274 104007                       ERROR    7                    ;REPORT ERROR
(2) 014276 013737 001164 001126 67$:   MOV      $TMP3,$BDDAT          ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 014304 013737 001220 001226        MOV      PORTB,PTNBR          ;CHANGE PORT NUMBER
(2) 014312 042737 100000 001164        BIC      #ATA,$TMP3           ;DON'T CHECK THE ATTN BIT
(2) 014320 023737 001124 001164        CMP      $GDDAT,$TMP3         ;SEE IF READ OK FROM PORT B.
(2) 014326 001401                       BEQ      68$                  ;BR IF OK
(2) 014330 104007                       ERROR    7                    ;REPORT ERROR
(2) 014332 001242 001126 001126 68$:   NOP
(2)
(2)                                     ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 014334 105737 001103                TSTB    $ERFLG                ;DID AN ERROR OCCUR ?
(4) 014340 001412                       BEQ      TST11                ;BR IF NOT
(2) 014342 032737 00 000 177570        BIT      #SW09,$SWR            ;SEE IF LOOP ON ERROR SET (SWR9=1)

```

```

(4) 014350 001406          BEQ     TST11          ;;BR IF NOT
(2) 014352 105037 001103    CLRB   SERFLG         ;;CLEAR THE ERROR FLAG
(2) 014356 005037 001170    CLR    $TIMES         ;;CLEAR THE MAX ITERATION COUNT
(2) 014362 000177 164522    JMP    $SLPERR        ;;GO TO THE LOOP ADDRESS
5325
5326
(3) *****
(3) *TEST 11          TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
(4) *
(4) *TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
(4) *
(4) *   A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
(4) *      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
(4) *
(3) *****
(3) TST11:
(3) 014366 000004          SCOPE          ;;INITIALIZE THE SCOPE HANDLER
(3) 014366 005737 001266    TST     KYBCTL       ;;PERFORMING ONLY SINGLE TESTS ?
(3) 014370 001406          BEQ     2$           ;;BR IF NOT
(3) 014374 100002          BPL     1$           ;;BR IF JUST ENTERED TEST
(3) 014400 000137 002110    JMP     EXEC         ;;RETURN & GET NEXT TEST NUMBER
(3) 014404 012737 177777 001266    1$:  MOV    #-1,KYBCTL ;;SET SINGLE TEST INDICATOR
(3) 014412 112737 000111 001102    2$:  MOVB   #1,$STNM   ;;TEST NUMBER
(3) 014420 012737 014442 001106    MOV    #TEST11,$LPADR ;;LOAD LOOP ON TEST ADDRESS
(3) 014426 012737 014442 001110    MOV    #TEST11,$LPERR ;;LOAD LOOP ON ERROR ADDRESS
(1) 014434 012737 000144 001170    MOV    #100,$TIMES   ;;DO 100. ITERATIONS
5327
(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) TEST11:
(2) 014442 113760 001220 000010    MOVB   PORTB,RHCS2(RO) ;;SELECT PORT B
(2) 014450 013737 001220 001226    MOV    PORTB,PTNBR    ;;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 014456 013737 001220 001230    MOV    PORTB,SEIZPT   ;;ADDR OF PORT WHICH WILL ISSUE RELEASE
(1)
(2) *****
(1) ;ISSUE A RELEASE COMMAND
(1)
(1) 014464 012760 000113 000000    MOV    #13,RHCS1(RO) ;;ISSUE A RELEASE COMMAND
(1)
(2) *****
(1) ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
(1)
(2)
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 014472 005037 001242          CLR     RELERR       ;;CLEAR THE 'RELEASE ERROR ' INDICATOR
(2) 014476 012737 000112 001122    MOV    #RHDS1,$BDADR  ;;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 014504 060037 001122          ADD    RO,$BDADR     ;;ADD THE I/O BASE ADDRESS
(2) 014510 012737 011700 001124    MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;;COMPARISON CONSTANT
(2) 014516 113760 001216 000010    MOVB   PORTA,RHCS2(RO) ;;SELECT PORT 'A'.
(2) 014524 015037 000012 001162    MOV    RHDS1(RO),$TMP2 ;;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 014532 013737 001162 001156    MOV    $TMP2,$TMP0    ;;COPY IT INTO '$TMP0'
(2) 014540 042737 100100 001156    BIC    #ATA!VV,$TMP0  ;;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 014546 113760 001220 000010    MOVB   PORTB,RHCS2(RO) ;;SELECT PORT B.
    
```

```

(2) 014554 016037 00002 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 014562 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(2) 014570 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 014576 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 014604 001006 BNE 64$ ;BR IF NOT
(2) 014606 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 014612 001037 BNE 66$ ;BR IF NOT
(2) 014614 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 014616 000137 015002 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(2) 014622 013737 001162 001126 64$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 014630 013737 001200 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014636 113760 001200 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 014644 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 014650 001414 BEQ 65$ ;BR IF ZERO
(2) 014652 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014660 013737 001164 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 014666 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 014674 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 014700 001004 BNE 66$ ;BR IF NOT
(2) 014702 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 014710 104030 ERROR 30 ;TYPE ERROR MESSAGE 30
(2) 014712 013737 001162 001126 66$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 014720 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 014726 042737 100001 001162 BIC #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
(2) 014734 023737 001121 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(2) 014742 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(2) 014744 104007 ERROR 7 ;REPORT ERROR
(2) 014746 013737 001164 001126 67$: MOV STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 014754 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 014762 042737 100001 001164 BIC #ATA,STMP3 ;DON'T CHECK THE ATTN BIT
(2) 014770 023737 001124 001164 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(2) 014776 001401 BEQ 68$ ;BR IF OK
(2) 015000 104007 ERROR 7 ;REPORT ERROR
(2) 015002 000240 68$: NOP
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2)
(2) 015004 105737 001101 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 015010 001412 BEQ TST12 ;BR IF NOT
(2) 015012 032737 001001 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 015020 001401 BEQ TST12 ;BR IF NOT
(2) 015022 005037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 015026 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 015032 000177 164052 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

5328  
5350  
5351

```

*****
*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 RHDS1 THROUGH PORT 'A'.
* VERIFY THAT THE DRIVE HAS BEEN SEIZED.

```



```

(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3) 015036
(3) 015036 000004
(3) 015040 005737 001266
(3) 015044 001406
(3) 015046 100002
(3) 015050 000137 002410
(3) 015054 012737 177777 001266
(3) 015062 112737 000012 001102
(3) 015070 012737 015112 001106
(3) 015076 012737 015112 001110
(1) 015104 012737 007640 001170
5352
5387
(3)
(2)
(2) 015112
(1)
(2)
(2)
(2)
(2) 015112 113760 001216 000010
(2) 015120 013737 001216 001230
(2) 015126 005060 000012
(3) 015132 113760 001220 000010
(3) 015140 013737 001220 001226
(2) 015146 013737 001220 001232
(2) 015154 016037 000012 001126
(2) 015162 010037 001122
(2) 015166 062737 000012 001122
(2) 015174 005037 001124
(2) 015200 023737 001124 001126
(2) 015206 001403
(2) 015210 104004
(2) 015212 000137 016406
(3) 015216 113760 001216 000010
(3) 015224 013737 001216 001226
(2) 015232 016037 000012 001126
(2) 015240 012737 011700 001124
(2) 015246 013737 001124 001160
(2) 015254 005137 001160
(2) 015260 013737 001126 001156

```

```

:
:
: 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.
:
:
:*****
:TEST12:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2S ;BR IF NOT
BPL 1S ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2S: MOV# 12,$STNM ;TEST NUMBER
MOV #TEST12,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000.,$TIMES ;DO 4000. ITERATIONS
:
:*****
:END OF 'SCOPE' SETUP - START OF MAIN TEST
:
TEST12:
:
:*****
;SEIZE THE DRIVE THROUGH PORT A
MOV# PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RHDS1(RO) ;WRITE RHDS1
MOV# PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RHDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BDAOR ;RH11 BASE ADDRESS
ADD #RHDS1,$BDAOR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
BEQ .+10 ;BR IF IT IS
ERROR 4 ;REPORT THE ERROR
JMP 1S ;BYPASS REST OF THE SUBTEST
MOV# PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RHDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS

```

```

(2) 015266 043737 001160 001156      BIC      $TMP1,$TMP0      ;CLEAR UNWANTED BITS
(2) 015274 023737 001124 001156      CMP      $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
(2) 015302 001401                      BEQ      .+4              ;BR IF THEY ARE
(2) 015304 104005                      ERROR 5                   ;REPORT THE ERROR
(1)
(2)                                     ;*****
(1)                                     ;DRIVE CLEAR THROUGH PORT A FIRST
(1)
(1) 015306 012760 000011 000000      MOV      #11,RHCS1(RO)   ;ISSUE DRIVE CLEAR THROUGH PORT A
(1)
(2)                                     ;*****
(1)                                     ;VERIFY THAT DRIVE STILL SEIZED BY PORT A
(1)
(2) 015314 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
(2) 015322 013737 001220 001226      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015330 005037 001236                      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015334 016037 000012 001126      MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 015342 012737 000012 001122      MOV      #RHDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015350 060037 001122                      ADD     RO,SBADR        ;ADD RH11 BASE ADDRESS
(2) 015354 005037 001124                      CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
(2) 015360 013737 001126 001156      MOV      SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 015366 042737 100000 001156      BIC      #1C7777,$TMP0  ;SAVE SPECIFIED BITS
(2) 015374 023737 001124 001156      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 015402 001414                      BEQ      64$            ;BR IF OK
(2) 015404 013737 001126 001166      MOV      SBDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 015412 042737 077777 001166      BIC      #77777,$TMP4   ;CLEAR THE MASKED BITS
(2) 015420 053737 001166 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015426 104033                      ERROR 33                ;TYPE MESSAGE 33
(2) 015430 005137 001236                      COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015434 000240                      NOP
(2) 015436 113760 001216 000010      64$: MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 015444 013737 001216 001226      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015452 005037 001236                      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015456 016037 000012 001126      MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 015464 012737 000012 001122      MOV      #RHDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015472 060037 001122                      ADD     RO,SBADR        ;ADD RH11 BASE ADDRESS
(2) 015476 012737 011700 001124      MOV      #M0L!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 015504 013737 001126 001156      MOV      SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 015512 042737 100000 001156      BIC      #1C7777,$TMP0  ;SAVE SPECIFIED BITS
(2) 015520 023737 001124 001156      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 015526 001414                      BEQ      65$            ;BR IF OK
(2) 015530 013737 001126 001166      MOV      SBDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 015536 042737 077777 001166      BIC      #77777,$TMP4   ;CLEAR THE MASKED BITS
(2) 015544 053737 001166 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015552 104033                      ERROR 33                ;TYPE MESSAGE 33
(2) 015554 005137 001236                      COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015560 000240                      NOP
(1)
(2)                                     ;*****
(1)                                     ;NOW ISSUE MASSBUS INIT
(1)
(1) 015562 012760 000040 000010      MOV      #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT
(1)
(2)                                     ;*****
    
```

```

(1)                                     ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
(1)
(2) 015570 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(2) 015576 013737 001220 001226      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015604 005037 001236               CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015610 016037 000012 001126      MOV   RHDS1(RO),SBDADR ;GET CONTENTS OF RHDS1
(2) 015616 012737 000012 001122      MOV   #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015624 060037 001122               ADD   RO,SBDADR ;ADD RH11 BASE ADDRESS
(2) 015630 005037 001124               CLR   $GDDAT ;WHAT REGISTER SHOULD BE
(2) 015634 013737 001126 001156      MOV   SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015642 042737 100000 001156      BIC   #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 015650 023737 001124 001156      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 015656 001414                       BEQ   66$ ;BR IF OK
(2) 015660 013737 001126 001166      MOV   SBDAT,$TMP4 ;COPY 'BAD DATA'
(2) 015666 042737 077777 001166      BIC   #77777,$TMP4 ;CLEAR THE MASKED BITS
(2) 015674 053737 001166 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015702 104034                       ERROR 34 ;TYPE MESSAGE 34
(2) 015704 005137 001236               COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015710 000240                       NOP
(2) 015712 113760 001216 000010      66$: MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(2) 015720 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015726 005037 001236               CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015732 016037 000012 001126      MOV   RHDS1(RO),SBDADR ;GET CONTENTS OF RHDS1
(2) 015740 012737 000012 001122      MOV   #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015746 060037 001122               ADD   RO,SBDADR ;ADD RH11 BASE ADDRESS
(2) 015752 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 015760 013737 001126 001156      MOV   SBDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015766 042737 100000 001156      BIC   #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 015774 023737 001124 001156      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 016002 001414                       BEQ   67$ ;BR IF OK
(2) 016004 013737 001126 001166      MOV   SBDAT,$TMP4 ;COPY 'BAD DATA'
(2) 016012 042737 077777 001166      BIC   #77777,$TMP4 ;CLEAR THE MASKED BITS
(2) 016020 053737 001166 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 016026 104034                       ERROR 34 ;TYPE MESSAGE 34
(2) 016030 005137 001236               COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 016034 000240                       NOP
(2)                                     ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 016036 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(3) 016044 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 016052 012760 000013 000000      MOV   #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 016060 005037 001242               CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 016064 012737 000012 001122      MOV   #RHDS1,SBDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 016072 060037 001122               ADD   RO,SBDADR ;ADD THE I/O BASE ADDRESS
(3) 016076 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 016104 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 016112 016037 000012 001162      MOV   RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 016120 013737 001162 001156      MOV   $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 016126 042737 100100 001156      BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 016134 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
    
```

```

(3) 016142 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 016150 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 016156 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 016164 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 016172 001006 BNE 68$ ;BR IF NOT
(3) 016174 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 016200 001045 BNE 70$ ;BR IF NOT
(3) 016202 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 016204 000137 016404 JMP 72$ ;BYPASS THE REST OF THE CHECKS
(3) 016210 013737 001162 001126 68$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 016216 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 016224 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 016232 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 016236 001414 BEQ 69$ ;BR IF ZERO
(3) 016240 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 016246 013737 001164 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 016254 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 016262 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 016266 001012 BNE 70$ ;BR IF NOT
(3) 016270 012737 177777 001242 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 016276 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 016304 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 016312 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 016314 013737 001162 001126 70$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 016322 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 016330 042737 100000 001162 BIC #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
(3) 016336 023737 001124 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 016344 001401 BEQ 71$ ;BR IF OK FROM PORT A.
(3) 016346 104007 ERROR 7 ;REPORT ERROR
(3) 016350 013737 001164 001126 71$: MOV STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 016356 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 016364 042737 100000 001164 BIC #ATA,STMP3 ;DON'T CHECK THE ATTN BIT
(3) 016372 023737 001124 001164 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 016400 001401 BEQ 72$ ;BR IF OK
(3) 016402 104007 ERROR 7 ;REPORT ERROR
(3) 016404 000240 72$: NOP
(2) 016406 1$:

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 016406 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 016412 001412 BEQ TST13 ;:BR IF NOT
(2) 016414 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 016422 001406 BEQ TST13 ;:BR IF NOT
(2) 016424 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 016430 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 016434 000177 162450 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

5406  
5407

```

(3) *****
(4) *TEST 13 TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
(4) *
(4) *VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
(4) * PORT TO RELEASE THE DRIVE.
(4) *

```

```

(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3) 016440
(3) 016440 000004
(3) 016442 005737 001266
(3) 016446 001406
(3) 016450 100002
(3) 016452 000137 002410
(3) 016456 012737 177777 001266
(3) 016464 112737 000013 001102
(3) 016472 012737 016514 001106
(3) 016500 012737 016514 001110
(1) 016506 012737 007640 001170
5408
5409
(3)
(2)
(2)
(2) 016514
(1)
(2)
(2)
(2)
(2)
(2) 016514 113760 001220 000010
(2) 016522 013737 001220 001230
(2) 016530 005060 000012
(3) 016534 113760 001216 000010
(3) 016542 013737 001216 001226
(2) 016550 013737 001216 001232
(2) 016556 016037 000012 001126
(2) 016564 010037 001122
(2) 016570 062737 000012 001122
(2) 016576 005037 001124
(2) 016602 023737 001124 001126
(2) 016610 001403
(2) 016612 104004
(2) 016614 000137 020010
(3) 016620 113760 001220 000010
(3) 016626 013737 001220 001226
(2) 016634 016037 000012 001126
(2) 016642 012737 011700 001124
(2) 016650 013737 001124 001160

```

- ```

;* A. SEIZE THE DRIVE BY WRITING 0'S INTO RHDS1 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.

```

\*\*\*\*\*

TEST13:

```

SCOPE ;INITIALIZE THE SCOPE HANDLER
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

```

```

;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

TEST13:

;\*\*\*\*\*

;SEIZE THE DRIVE THROUGH PORT B

```

MOVb PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RHDS1(RO) ;WRITE RHDS1
MOVb PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RHDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$BADDR ;RH11 BASE ADDRESS
ADD #RHDS1,$BADDR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
BEQ .+10 ;BR IF IT IS
ERROR 4 ;REPORT THE ERROR
JMP 1$ ;BYPASS REST OF THE SUBTEST
MOVb PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RHDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK

```

```

(2) 016656 005137 001160          COM      STMP1          ;COMPLEMENT THE EXPECTED STATUS
(2) 016662 013737 001126 001156    MOV      $BDDAT,$STMP0 ;SAVE THE ACTUAL STATUS
(2) 016670 043737 001160 001156    BIC      STMP1,$STMP0  ;CLEAR UNWANTED BITS
(2) 016676 023737 001124 001156    CMP      $GDDAT,$STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 016704 001401          BEQ      .+4           ;BR IF THEY ARE
(2) 016706 104005          ERROR 5          ;REPORT THE ERROR
(1)
(2) ;*****
(1) ;DRIVE CLEAR THROUGH PORT B FIRST
(1)
(1) 016710 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT B
(1)
(2) ;*****
(1) ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
(1)
(2) 016716 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 016724 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 016732 005037 001236          CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 016736 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 016744 012737 000012 001122      MOV      #RHDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 016752 060037 001122          ADD      RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 016756 005037 001124          CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 016762 013737 001126 001156      MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 016770 042737 100000 001156      BIC      #1C7777,$STMP0 ;SAVE SPECIFIED BITS
(2) 016776 023737 001124 001156      CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
(2) 017004 001414          BEQ      64$ ;BR IF OK
(2) 017006 013737 001126 001166      MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
(2) 017014 042737 077777 001166      BIC      #77777,$STMP4 ;CLEAR THE MASKED BITS
(2) 017022 053737 001166 001124      BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017030 104033          ERROR 33 ;TYPE MESSAGE 33
(2) 017032 005137 001236          COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017036 000240          NOP
(2) 017040 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(2) 017046 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017054 005037 001236          CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 017060 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 017066 012737 000012 001122      MOV      #RHDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 017074 060037 001122          ADD      RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 017100 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 017106 013737 001126 001156      MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 017114 042737 100000 001156      BIC      #1C7777,$STMP0 ;SAVE SPECIFIED BITS
(2) 017122 023737 001124 001156      CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
(2) 017130 001414          BEQ      65$ ;BR IF OK
(2) 017132 013737 001126 001166      MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
(2) 017140 042737 077777 001166      BIC      #77777,$STMP4 ;CLEAR THE MASKED BITS
(2) 017146 053737 001166 001124      BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017154 104033          ERROR 33 ;TYPE MESSAGE 33
(2) 017156 005137 001236          COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017162 000240          NOP
(1)
(2) ;*****
(1) ;NOW ISSUE MASSBUS INIT
(1)
(1) 017164 012760 000040 000010      MOV      #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT
    
```

```

(1)
(2) ;:*****
(1) ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
(1)
(2) 017172 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 017200 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017206 005037 001236              CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 017212 016037 000012 001126      MOV    RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 017220 012737 000012 001122      MOV    #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 017226 060037 001122              ADD    RO, $BDADR ;ADD RH11 BASE ADDRESS
(2) 017232 005037 001124              CLR    $GDDAT ;WHAT REGISTER SHOULD BE
(2) 017236 013737 001126 001156      MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 017244 042737 100000 001156      BIC    #1C7777, $TMP0 ;SAVE SPECIFIED BITS
(2) 017252 023737 001124 001156      CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 017260 001414              BEQ    66$ ;BR IF OK
(2) 017262 013737 001126 001166      MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 017270 042737 077777 001166      BIC    #77777, $TMP4 ;CLEAR THE MASKED BITS
(2) 017276 053737 001166 001124      BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017304 104034              ERROR  34 ;TYPE MESSAGE 34
(2) 017306 005137 001236              COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017312 000240              NOP
(2) 017314 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 017322 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017330 005037 001236              CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 017334 016037 000012 001126      MOV    RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 017342 012737 000012 001122      MOV    #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 017350 060037 001122              ADD    RO, $BDADR ;ADD RH11 BASE ADDRESS
(2) 017354 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 017362 013737 001126 001156      MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 017370 042737 100000 001156      BIC    #1C7777, $TMP0 ;SAVE SPECIFIED BITS
(2) 017376 023737 001124 001156      CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 017404 001414              BEQ    67$ ;BR IF OK
(2) 017406 013737 001126 001166      MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 017414 042737 077777 001166      BIC    #77777, $TMP4 ;CLEAR THE MASKED BITS
(2) 017422 053737 001166 001124      BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017430 104034              ERROR  34 ;TYPE MESSAGE 34
(2) 017432 005137 001236              COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017436 000240              NOP
(2)
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 017440 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 017446 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017454 012760 000013 000000      MOV    #13, RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 017462 005037 001242              CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 017466 012737 000012 001122      MOV    #RHDS1, $BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 017474 060037 001122              ADD    RO, $BDADR ;ADD THE I/O BASE ADDRESS
(3) 017500 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
(3) 017506 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 017514 016037 000012 001162      MOV    RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 017522 013737 001162 001156      MOV    $TMP2, $TMP0 ;COPY IT INTO 'TMP0'

```

```

(3) 017530 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 017536 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 017544 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 017552 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO 'STMP1'
(3) 017560 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 017566 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 017574 001006 BNE 68$ ;BR IF NOT
(3) 017576 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 017602 001045 BNE 70$ ;BR IF NOT
(3) 017604 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 017606 000137 020006 JMP 72$ ;BYPASS THE REST OF THE CHECKS
(3) 017612 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 017620 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 017626 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 017634 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 017640 001414 BEQ 69$ ;BR IF ZERO
(3) 017642 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 017650 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 017656 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 017664 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 017670 001012 BNE 70$ ;BR IF NOT
(3) 017672 012737 177777 001242 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 017700 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 017706 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 017714 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 017716 013737 001162 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 017724 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 017732 042737 100000 001162 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(3) 017740 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 017746 001401 BEQ 71$ ;BR IF OK FROM PORT A.
(3) 017750 104007 ERROR 7 ;REPORT ERROR
(3) 017752 013737 001164 001126 71$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 017760 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 017766 042737 100000 001164 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 017774 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 020002 001401 BEQ 72$ ;BR IF OK
(3) 020004 104007 ERROR 7 ;REPORT ERROR
(3) 020006 000240 72$: NOP
(2) 020010 1$:

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 020010 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
(4) 020014 001412 BEQ TST14 ;:BR IF NOT
(2) 020016 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 020024 001406 BEQ TST14 ;:BR IF NOT
(2) 020026 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
(2) 020032 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 020036 000177 161046 JMP $JLPERR ;GO TO THE LOOP ADDRESS

```

5410  
5429  
5430  
(3)  
(4)

\*\*\*\*\*  
;\*TEST 14 TEST RESET ATTENTION 'A' BY MASSBUS CLEAR  
;\*



```

(4) ;*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
(4) ;* SEIZING PORT.
(4) ;*
(4) ;* A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
(4) ;* SET.
(4) ;*
(4) ;* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
(4) ;*
(4) ;* C. ISSUE A MASSBUS CLEAR.
(4) ;*
(4) ;*
(4) ;* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION
(4) ;* BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
(4) ;* 'B' IS STILL SET.
(4) ;*
(4) ;*

```

```

(3) *****
(3) TST14:
(3) 020042 000004 SCOPE ; INITIALIZE THE SCOPE HANDLER
(3) 020044 005737 001266 TST KYBCTL ; PERFORMING ONLY SINGLE TESTS ?
(3) 020050 001406 BEQ 2$ ; BR IF NOT
(3) 020052 100002 BPL 1$ ; BR IF JUST ENTERED TEST
(3) 020054 000137 002410 JMP EXEC ; RETURN & GET NEXT TEST NUMBER
(3) 020060 012737 177777 001266 1$: MOV #-1,KYBCTL ; SET SINGLE TEST INDICATOR
(3) 020066 112737 000014 001102 2$: MOVB #14,$TSTNM ; TEST NUMBER
(3) 020074 012737 020116 001106 MOV #TEST14,$LPADR ; LOAD LOOP ON TEST ADDRESS
(3) 020102 012737 020116 001110 MOV #TEST14,$LPERR ; LOAD LOOP ON ERROR ADDRESS
(1) 020110 012737 000004 001170 MOV #4,$TIMES ; DO 4 ITERATIONS

```

```

5431
5472
(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2) ;*****

```

```

(2) 020116 TEST14:
(2) ;*****
(2) ;SET ATTENTION BITS FOR BOTH PORTS
(2) 020116 113760 001216 000010 MOVB PORTA,RHCS2(RO) ; SELECT PORT 'A'
(2) 020124 012760 177777 000014 MOV #-1,RHER1(RO) ; FORCE ERRORS
(2) 020132 005060 000014 CLR RHER1(RO) ; CLEAR THE ERRORS
(2) 020136 013760 001220 000010 MOV PORTB,RHCS2(RO) ; SELECT THE OTHER PORT
(2) 020144 005760 000012 TST RHDS1(RO) ; WAIT FOR DRIVE TO TIMEOUT
(2) 020150 001775 BEQ .-4 ; BR IF DRIVE HASN'T TIMED OUT
(2) 020152 012760 177777 000014 MOV #-1,RHER1(RO) ; FORCE ERRORS ON PORT 'B'
(2) 020160 005060 000014 CLR RHER1(RO) ; CLEAR THE ERRORS
(2) 020164 113760 001216 000010 MOVB PORTA,RHCS2(RO) ; SELECT PORT "A" AGAIN
(2) 020172 005760 000012 TST RHDS1(RO) ; WAIT FOR DRIVE TO TIMEOUT
(2) 020176 001775 BEQ .-4 ; BR IF DRIVE HASN'T TIMED OUT

```

```

(1) ;*****
(1) ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
(3) 020200 113760 001216 000010 MOVB PORTA,RHCS2(RO) ; SELECT PORT A
(3) 020206 013737 001216 001226 MOV PORTA,PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

(3) 020214 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 020220 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(3) 020226 012737 000012 001122 MOV #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 020234 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
(3) 020240 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
(3) 020246 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(3) 020254 042737 077777 001156 BIC #↑CATA, $TMP0 ;SAVE SPECIFIED BITS
(3) 020262 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(3) 020270 001414 BEQ 64$ ;BR IF OK
(3) 020272 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(3) 020300 042737 100000 001166 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
(3) 020306 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 020314 104010 ERROR 10 ;REPORT THE ERROR
(3) 020316 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 020322 000240 64$: NOP
(2) 020324 005737 001236 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
(2) 020330 001402 BEQ +6 ;BR IF IT WAS
(2) 020332 000137 021340 JMP 1$ ;BYPASS REST OF TEST IF NOT
(3) 020336 113760 001220 000010 MOV# PORTB, RHCS2(RO) ;SELECT PORT B
(3) 020344 013737 001220 001226 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 020352 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 020356 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(3) 020364 012737 000012 001122 MOV #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 020372 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
(3) 020376 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
(3) 020404 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(3) 020412 042737 077777 001156 BIC #↑CATA, $TMP0 ;SAVE SPECIFIED BITS
(3) 020420 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(3) 020426 001414 BEQ 65$ ;BR IF OK
(3) 020430 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(3) 020436 042737 100000 001166 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
(3) 020444 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 020452 104010 ERROR 10 ;REPORT THE ERROR
(3) 020454 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 020460 000240 65$: NOP
(2) 020462 005737 001236 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
(2) 020466 001402 BEQ +6 ;BR IF IT WAS
(2) 020470 000137 021340 JMP 1$ ;BYPASS REST OF TEST IF NOT
(1)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT A
(2) MOV# PORTA, RHCS2(RO) ;SELECT PORT A
(2) 020474 113760 001216 000010 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 020502 013737 001216 001230 CLR RHDS1(RO) ;WRITE RHDS1
(2) 020510 005060 000012 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 020514 013737 001220 001232
(1)
(2) ;*****
(1) ;ISSUE MASSBUS INIT TO PORT A
(1) MOV #CLR, RHCS2(RO) ;MASSBUS INIT
(1) 020522 012760 000040 000010 MOV# PORTA, RHCS2(RO) ;SELECT PORT A AGAIN
(1) 020530 113760 001216 000010

```

```

(2) ;*****
(1) ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
(1)
(2) 020536 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 020542 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 020550 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 020556 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
(2) 020562 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 020566 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 020574 042737 077777 001156 BIC #CATA, $TMP0 ;SAVE SPECIFIED BITS
(2) 020602 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 020610 001414 BEQ 66$ ;BR IF OK
(2) 020612 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 020620 042737 100000 001166 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
(2) 020626 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 020634 104047 ERROR 47 ;TYPE MESSAGE 47
(2) 020636 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 020642 000240 66$: NOP

(1) ;*****
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 020644 113760 001216 000010 MOV#B PORTA, RHCS2(RO) ;SELECT PORT A
(3) 020652 013737 001216 001226 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 020660 012760 000013 000000 MOV #13, RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 020666 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 020672 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 020700 060037 001122 ADD RO, $B0ADR ;ADD THE I/O BASE ADDRESS
(3) 020704 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
(3) 020712 113760 001216 000010 MOV#B PORTA, RHCS2(RO) ;SELECT PORT A.
(3) 020720 016037 000012 001162 MOV RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 020726 013737 001162 001156 MOV $TMP2, $TMP0 ;COPY IT INTO 'TMP0'
(3) 020734 042737 100100 001156 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 020742 113760 001220 000010 MOV#B PORTB, RHCS2(RO) ;SELECT PORT B.
(3) 020750 016037 000012 001164 MOV RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 020756 013737 001164 001160 MOV $TMP3, $TMP1 ;COPY IT INTO 'TMP1'
(3) 020764 042737 100100 001160 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 020772 023737 001156 001160 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 021000 001006 BNE 67$ ;BR IF NOT
(3) 021002 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 021006 001045 BNE 69$ ;BR IF NOT
(3) 021010 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 021012 000137 021212 JMP 71$ ;BYPASS THE REST OF THE CHECKS
(3) 021016 013737 001162 001126 67$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 021024 013737 001220 001226 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 021032 113760 001220 000010 MOV#B PORTB, RHCS2(RO) ;SELECT PORT B.
(3) 021040 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 021044 001414 BEQ 68$ ;BR IF ZERO
(3) 021046 013737 001216 001226 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 021054 013737 001164 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

```

(3) 021062 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 021070 005737 001160             TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 021074 001012                     BNE    69$            ;BR IF NOT
(3) 021076 012737 177777 001242 68$:  MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 021104 012760 000011 000000      MOV    #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 021112 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 021120 104026                     ERROR  26             ;TYPE ERROR MESSAGE 26
(3) 021122 013737 001162 001126 69$:  MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 021130 013737 001216 001226      MOV    PORTA,PTNBR    ;CHANGE PORT NUMBER
(3) 021136 042737 100000 001162      BIC    #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
(3) 021144 023737 001124 001162      CMP    $GDDAT,$TMP2  ;ALL BITS OK ?
(3) 021152 001401                     BEQ    70$            ;BR IF OK FROM PORT A.
(3) 021154 104007                     ERROR  7              ;REPORT ERROR
(3) 021156 013737 001164 001126 70$:  MOV    $TMP3,$BDDAT   ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 021164 013737 001220 001226      MOV    PORTB,PTNBR   ;CHANGE PORT NUMBER
(3) 021172 042737 100000 001164      BIC    #ATA,$TMP3     ;DON'T CHECK THE ATTN BIT
(3) 021200 023737 001124 001164      CMP    $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.
(3) 021206 001401                     BEQ    71$            ;BR IF OK
(3) 021210 104007                     ERROR  7              ;REPORT ERROR
(3) 021212 000240 71$:  NOP

```

```

(1)
(2) ;*****
(1) ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)
(1)

```

```

(2) 021214 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 021222 013737 001220 001226      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 021230 005037 001236             CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 021234 016037 000012 001126      MOV    RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 021242 012737 000012 001122      MOV    #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 021250 060037 001122             ADD    RO,$BDDADR    ;ADD RH11 BASE ADDRESS
(2) 021254 012737 100000 001124      MOV    #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
(2) 021262 013737 001126 001156      MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 021270 042737 077777 001156      BIC    #ICATA,$TMP0  ;SAVE SPECIFIED BITS
(2) 021276 023737 001124 001156      CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
(2) 021304 001414                     BEQ    72$            ;BR IF OK
(2) 021306 013737 001126 001166      MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
(2) 021314 042737 100000 001166      BIC    #ATA,$TMP4    ;CLEAR THE MASKED BITS
(2) 021322 053737 001166 001124      BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 021330 104050                     ERROR  50             ;TYPE MESSAGE 50
(2) 021332 005137 001236             COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 021336 000240 72$:  NOP
(2) 1$:

```

```

(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 021340 105737 001103             TSTB   $ERFLG         ;DID AN ERROR OCCUR ?
(4) 021344 001412                     BEQ    TST15          ;BR IF NOT
(2) 021346 032737 001000 177570      BIT    #SW09,SWR      ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 021354 001406                     BEQ    TST15          ;BR IF NOT
(2) 021356 105037 001103             CLRB   $ERFLG         ;CLEAR THE ERROR FLAG
(2) 021362 005037 001170             CLR    $TIMES         ;CLEAR THE MAX ITERATION COUNT
(2) 021366 000177 157516             JMP    $SLPERR        ;GO TO THE LOOP ADDRESS

```

5490  
5491

;\*\*\*\*\*

(3)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(3)  
(3) 021372  
(3) 021372 000004  
(3) 021374 005737 001266  
(3) 021400 001406  
(3) 021402 100002  
(3) 021404 000137 002410  
(3) 021410 012737 177777 001266  
(3) 021416 112737 000015 001102  
(3) 021424 012737 021446 001106  
(3) 021432 012737 021446 001110  
(1) 021440 012737 000004 001170  
5492  
5493  
(3)  
(2)  
(2)  
(2) 021446  
(2)  
(2)  
(2)  
(2)  
(2) 021446 113760 001216 000010  
(2) 021454 012760 177777 000014  
(2) 021462 005060 000014  
(2) 021466 013760 001220 000010  
(2) 021474 005760 000012  
(2) 021500 001775  
(2) 021502 012760 177777 000014  
(2) 021510 005060 000014  
(2) 021514 113760 001216 000010  
(2) 021522 005760 000012  
(2) 021526 001775  
(1)  
(2)  
(1)  
(1)  
(3) 021530 113760 001220 000010

```

*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 RHDS1.
*
* C. ISSUE A MASSBUS CLEAR.
*
* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION
* BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
* 'A' IS STILL SET.
*
*****
TST15:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
MOV #15,$STSTM ;TEST NUMBER
MOV #TEST15,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST15,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST15:
*****
;SET ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ;SELECT PORT 'A'
MOV #-1,RHER1(RO) ;FORCE ERRORS
CLR RHER1(RO) ;CLEAR THE ERRORS
MOV PORTB,RHCS2(RO) ;SELECT THE OTHER PORT
TST RHDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ -4 ;BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RHER1(RO) ;FORCE ERRORS ON PORT 'B'
CLR RHER1(RO) ;CLEAR THE ERRORS
MOVB PORTA,RHCS2(RO) ;SELECT PORT "A" AGAIN
TST RHDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
BEQ -4 ;BR IF DRIVE HASN'T TIMED OUT

*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTB,RHCS2(RO) ;SELECT PORT B

```

```

(3) 021536 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 021544 005037 001236              CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 021550 016037 000012 001126      MOV    RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
(3) 021556 012737 000012 001122      MOV    #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 021564 060037 001122              ADD    RO,SBDADR   ;ADD RH11 BASE ADDRESS
(3) 021570 012737 100000 001124      MOV    #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
(3) 021576 013737 001126 001156      MOV    SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(3) 021604 042737 077777 001156      BIC    #ICATA,STMP0 ;SAVE SPECIFIED BITS
(3) 021612 023737 001124 001156      CMP    SGDDAT,STMP0 ;COMPARE THE BITS
(3) 021620 001414                      BEQ    64$        ;BR IF OK
(3) 021622 013737 001126 001166      MOV    SBDAT,STMP4 ;COPY 'BAD DATA'
(3) 021630 042737 100000 001166      BIC    #ATA,STMP4  ;CLEAR THE MASKED BITS
(3) 021636 053737 001166 001124      BIS    STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 021644 104010                      ERROR  10        ;REPORT THE ERROR
(3) 021646 005137 001236              COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 021652 000240                      NOP
(2) 021654 005737 001236      64$:  TST    CKERR      ;WAS ATTN BIT FOR PORT B SET ?
(2) 021660 001402                      BEQ    .+6        ;BR IF IT WAS
(2) 021662 000137 022670                      JMP    IS         ;BYPASS REST OF TEST IF NOT
(3) 021666 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(3) 021674 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 021702 005037 001236              CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 021706 016037 000012 001126      MOV    RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
(3) 021714 012737 000012 001122      MOV    #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 021722 060037 001122              ADD    RO,SBDADR   ;ADD RH11 BASE ADDRESS
(3) 021726 012737 100000 001124      MOV    #ATA,SGDDAT ;WHAT REGISTER SHOULD BE
(3) 021734 013737 001126 001156      MOV    SBDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(3) 021742 042737 077777 001156      BIC    #ICATA,STMP0 ;SAVE SPECIFIED BITS
(3) 021750 023737 001124 001156      CMP    SGDDAT,STMP0 ;COMPARE THE BITS
(3) 021756 001414                      BEQ    65$        ;BR IF OK
(3) 021760 013737 001126 001166      MOV    SBDAT,STMP4 ;COPY 'BAD DATA'
(3) 021766 042737 100000 001166      BIC    #ATA,STMP4  ;CLEAR THE MASKED BITS
(3) 021774 053737 001166 001124      BIS    STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 022002 104010                      ERROR  10        ;REPORT THE ERROR
(3) 022004 005137 001236              COM    CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 022010 000240                      NOP
(2) 022012 005737 001236      65$:  TST    CKERR      ;WAS ATTN BIT FOR PORT A SET ?
(2) 022016 001402                      BEQ    .+6        ;BR IF IT WAS
(2) 022020 000137 022670                      JMP    IS         ;BYPASS REST OF TEST IF NOT
(1)
(2)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) 022024 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 022032 013737 001220 001230      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 022040 005060 000012              CLR    RHDS1(RO)   ;WRITE RHDS1
(2) 022044 013737 001216 001232      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(1)
(2) ;*****
(1) ;ISSUE MASSBUS INIT TO PORT B
(1) 022052 012760 000040 000010      MOV    #CLR,RHCS2(RO) ;MASSBUS INIT
(1) 022060 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B AGAIN

```

```

(1)
(2) ;:*****
(1) ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
(1)
(2) 022066 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 022072 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 022100 012737 000012 001122 MOV #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 022106 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 022112 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
(2) 022116 013737 001126 001156 MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 022124 042737 077777 001156 BIC #ICATA,STMP0 ;SAVE SPECIFIED BITS
(2) 022132 023737 001124 001156 CMP SGDDAT,STMP0 ;COMPARE THE BITS
(2) 022140 001414 BEQ 66$ ;BR IF OK
(2) 022142 013737 001126 001165 MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
(2) 022150 042737 100000 001166 BIC #ATA,STMP4 ;CLEAR THE MASKED BITS
(2) 022156 053737 001166 001124 BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 022164 104047 ERROR 47 ;TYPE MESSAGE 47
(2) 022166 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 022172 000240 66$: NOP

(1) ;:*****
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 022174 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 022202 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 022210 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B

(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 022216 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 022222 012737 000012 001122 MOV #RHDS1,SBADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 022230 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
(3) 022234 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
(3) 022242 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 022250 016037 000012 001162 MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 022256 013737 001162 001156 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 022264 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 022272 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 022300 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 022306 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 022314 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 022322 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 022330 001006 BNE 67$ ;BR IF NOT
(3) 022332 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 022336 001045 BNE 69$ ;BR IF NOT
(3) 022340 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 022342 000137 022542 JMP 71$ ;BYPASS THE REST OF THE CHECKS
(3) 022346 013737 001162 001126 67$: MOV STMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 022354 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 022362 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 022370 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 022374 001414 BEQ 68$ ;BR IF ZERO
(3) 022376 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

(3) 022404 013737 001164 001126      MOV      $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
(3) 022412 113760 001216 000010      MOVVB   PORTA,RHCS2(RO)   ;SELECT PORT A.
(3) 022420 005737 001160              TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 022424 001012              BNE     69$              ;BR IF NOT
(3) 022426 012737 177777 001242 68$:  MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
(3) 022434 012760 000011 000000      MOV     #11,RHCS1(RO)    ;CLEAR THE DRIVE
(3) 022442 012760 000013 000000      MOV     #13,RHCS1(RO)    ;RELEASE THE DRIVE
(3) 022450 104026              ERROR   26              ;TYPE ERROR MESSAGE 26
(3) 022452 013737 001162 001126 69$:  MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 022460 013737 001216 001226      MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
(3) 022466 042737 100000 001162      BIC     #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
(3) 022474 023737 001124 001162      CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
(3) 022502 001401              BEQ     70$              ;BR IF OK FROM PORT A.
(3) 022504 104007              ERROR   7               ;REPORT ERROR
(3) 022506 013737 001164 001126 70$:  MOV     $TMP3,$BDDAT     ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 022514 013737 001220 001226      MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
(3) 022522 042737 100000 001164      BIC     #ATA,$TMP3       ;DON'T CHECK THE ATTN BIT
(3) 022530 023737 001124 001164      CMP     $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
(3) 022536 001401              BEQ     71$              ;BR IF OK
(3) 022540 104007              ERROR   7               ;REPORT ERROR
(3) 022542 000240 71$:  NOP

(1)
(2)
(1)
(1)
(2) 022544 113760 001216 000010      MOVVB   PORTA,RHCS2(RO)   ;SELECT PORT A
(2) 022552 013737 001216 001226      MOV     PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 022560 005037 001236              CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 022564 016037 000012 001126      MOV     RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 022572 012737 000012 001122      MOV     #RHDS1,$BADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 022600 060037 001122              ADD     RO,$BADR         ;ADD RH11 BASE ADDRESS
(2) 022604 012737 100000 001124      MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
(2) 022612 013737 001126 001156      MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 022620 042737 077777 001156      BIC     #ICATA,$TMP0     ;SAVE SPECIFIED BITS
(2) 022626 023737 001124 001156      CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
(2) 022634 001414              BEQ     72$              ;BR IF OK
(2) 022636 013737 001126 001166      MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
(2) 022644 042737 100000 001166      BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
(2) 022652 053737 001166 001124      BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 022660 104050              ERROR   50              ;TYPE MESSAGE 50
(2) 022662 005137 001236              COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 022666 000240 72$:  NOP
(2) 022670 15$:

(2)
(2)
(2)
(2)
(2) 022670 105737 001103              TSTB   SERFLG           ;DID AN ERROR OCCUR ?
(4) 022674 001412              BEQ     TST16            ;BR IF NOT
(2) 022676 032737 001000 177570      BIT     #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 022704 001406              BEQ     TST16            ;BR IF NOT
(2) 022706 105037 001103              CLRB   SERFLG           ;CLEAR THE ERROR FLAG
(2) 022712 005037 001170              CLR    $TIMES           ;CLEAR THE MAX ITERATION COUNT
(2) 022716 000177 156166              JMP    $JLPERP          ;GO TO THE LOOP ADDRESS

```



5498  
5512  
5513  
(3)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(3)

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

(3) 022722  
(3) 022722 000004  
(3) 022724 005737 001266  
(3) 022730 001406  
(3) 022732 100002  
(3) 022734 000137 002410  
(3) 022740 012737 177777 001266  
(3) 022746 112737 000016 001102  
(3) 022754 012737 022776 001106  
(3) 022762 012737 022776 001110  
(1) 022770 012737 000004 001170

```
TST16:
SCOPE ; INITIALIZE THE SCOPE HANDLER
TST KYBCTL ; PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ; BR IF NOT
BPL 15 ; BR IF JUST ENTERED TEST
JMP EXEC ; RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ; SET SINGLE TEST INDICATOR
2$: MOVB #16,$TSTNM ; TEST NUMBER
MOV #TEST16,$LPADR ; LOAD LOOP ON TEST ADDRESS
MOV #TEST16,$LPERR ; LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ; DO 4 ITERATIONS
```

5514  
5550

(3)  
(2)  
(2)  
(2) 022776  
(2)  
(2)  
(2)  
(2) 022776 113760 001216 000010  
(2) 023004 012760 177777 000014  
(2) 023012 005060 000014  
(2) 023016 013760 001220 000010  
(2) 023024 005760 000012  
(2) 023030 001775  
(2) 023032 012760 177777 000014  
(2) 023040 005060 000014  
(2) 023044 113760 001216 000010  
(2) 023052 005760 000012  
(2) 023056 001775  
(1)  
(2)  
(1)  
(1)  
(3) 023060 113760 001216 000010  
(3) 023066 013737 001216 001226

```
*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST16:
*****
;SET ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ; SELECT PORT 'A'
MOV #-1,RHER1(RO) ; FORCE ERRORS
CLR RHER1(RO) ; CLEAR THE ERRORS
MOV PORTB,RHCS2(RO) ; SELECT THE OTHER PORT
TST RHDS1(RO) ; WAIT FOR DRIVE TO TIMEOUT
BEQ -4 ; BR IF DRIVE HASN'T TIMED OUT
MOV #-1,RHER1(RO) ; FORCE ERRORS ON PORT 'B'
CLR RHER1(RO) ; CLEAR THE ERRORS
MOVB PORTA,RHCS2(RO) ; SELECT PORT "A" AGAIN
TST RHDS1(RO) ; WAIT FOR DRIVE TO TIMEOUT
BEQ -4 ; BR IF DRIVE HASN'T TIMED OUT
*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
MOVB PORTA,RHCS2(RO) ; SELECT PORT A
MOV PORTA,PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```

(3) 023074 005037 001236          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 023100 016037 000012 001126  MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(3) 023106 012737 000012 001122  MOV      #RHDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 023114 060037 001122          ADD      RO, $BDADR      ;ADD RH11 BASE ADDRESS
(3) 023120 012737 100000 001124  MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
(3) 023126 013737 001126 001156  MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'STMP0'
(3) 023134 042737 077777 001156  BIC      #1CATA, $TMP0   ;SAVE SPECIFIED BITS
(3) 023142 023737 001124 001156  CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
(3) 023150 001414          BEQ      64$            ;BR IF OK
(3) 023152 013737 001126 001166  MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
(3) 023160 042737 100000 001166  BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
(3) 023166 053737 001166 001124  BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 023174 104010          ERROR    10             ;REPORT THE ERROR
(3) 023176 005137 001236          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 023202 000240          NOP
(2) 023204 005737 001236          TST      CKERR          ;WAS ATTN BIT FOR PORT A SET ?
(2) 023210 001402          BEQ      .+6           ;BR IF IT WAS
(2) 023212 000137 024156          JMP      1$            ;BYPASS REST OF TEST IF NOT
(3) 023216 113760 001220 000010  MOVVB   PORTB, RHCS2(RO) ;SELECT PORT B
(3) 023224 013737 001220 001226  MOV      PORTB, $PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 023232 005037 001236          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 023236 016037 000012 001126  MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(3) 023244 012737 000012 001122  MOV      #RHDS1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 023252 060037 001122          ADD      RO, $BDADR      ;ADD RH11 BASE ADDRESS
(3) 023256 012737 100000 001124  MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
(3) 023264 013737 001126 001156  MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'STMP0'
(3) 023272 042737 077777 001156  BIC      #1CATA, $TMP0   ;SAVE SPECIFIED BITS
(3) 023300 023737 001124 001156  CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
(3) 023306 001414          BEQ      65$            ;BR IF OK
(3) 023310 013737 001126 001166  MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
(3) 023316 042737 100000 001166  BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
(3) 023324 053737 001166 001124  BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 023332 104010          ERROR    10             ;REPORT THE ERROR
(3) 023334 005137 001236          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 023340 000240          NOP
(2) 023342 005737 001236          TST      CKERR          ;WAS ATTN BIT FOR PORT B SET ?
(2) 023346 001402          BEQ      .+6           ;BR IF IT WAS
(2) 023350 000137 024156          JMP      1$            ;BYPASS REST OF TEST IF NOT
(1)
(2)
(2)
(2)
(2)
(2)
(2) 023354 005037 001242          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 023360 012737 000012 001122  MOV      #RHDS1, $BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 023366 060037 001122          ADD      RO, $BDADR      ;ADD THE I/O BASE ADDRESS
(2) 023372 012737 111700 001124  MOV      #11700, $GDDAT  ;COMPARISON CONSTANT
(2) 023400 113760 001216 000010  MOVVB   PORTA, RHCS2(RO) ;SELECT PORT A.
(2) 023406 016037 000012 001162  MOV      RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 023414 013737 001162 001156  MOV      $TMP2, $TMP0    ;COPY IT INTO 'STMP0'
(2) 023422 042737 100100 001156  BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 023430 113760 001220 000010  MOVVB   PORTB, RHCS2(RO) ;SELECT PORT B.
(2) 023436 016037 000012 001164  MOV      RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 023444 013737 001164 001160  MOV      $TMP3, $TMP1   ;COPY IT INTO 'STMP1'
;*****
;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

```

(2) 023452 042737 100100 001160 BIC #ATA:VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 023460 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 023466 001006 BNE 66$ ;BR IF NOT
(2) 023470 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 023474 001045 BNE 68$ ;BR IF NOT
(2) 023476 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 023500 000137 023664 JMP 70$ ;BYPASS THE REST OF THE CHECKS
(2) 023504 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 023512 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 023520 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 023526 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 023532 001414 BEQ 67$ ;BR IF ZERO
(2) 023534 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 023542 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 023550 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 023556 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 023562 001012 BNE 68$ ;BR IF NOT
(2) 023564 012737 177777 001242 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 023572 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(2) 023600 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 023606 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(2) 023610 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 023616 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 023624 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 023632 001401 BEQ 69$ ;BR IF OK FROM PORT A.
(2) 023634 104007 ERROR 7 ;REPORT ERROR
(2) 023636 013737 001164 001126 69$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 023644 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 023652 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 023660 001401 BEQ 70$ ;BR IF OK
(2) 023662 104007 ERROR 7 ;REPORT ERROR
(2) 023664 000240 70$: NOP
(1) 023666 005737 001242 TST RELERR ;WAS DRIVE IN NEUTRAL ?
(1) 023672 001402 BEQ +6 ;BR IF IT WAS
(1) 023674 000137 024156 JMP IS ;BYPASS RESET OF TEST
(2) ;*****
(1) ;ISSUE THE MASSBUS INIT
(1)
(1) 023700 012760 000040 000010 MOV #CLR,RHCS2(RO) ;ISSUE A MASSBUS INIT
(2) ;*****
(1) ;CHECK THE ATTENTION BITS OF BOTH PORTS
(1)
(2) 023706 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 023714 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 023722 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 023726 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 023734 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 023742 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 023746 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 023754 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 023762 042737 077777 001156 BIC #+CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 023770 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 023776 001414 BEQ 71$ ;BR IF OK
    
```



```

(3) 024226 012737 177777 001266 1S: MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 024234 112737 000017 001102 2S: MOVB #17,$STNM ;TEST NUMBER
(3) 024242 012737 024264 001106 MOV #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 024250 012737 024264 001110 MOV #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 024256 012737 007640 001170 MOV #4000,$TIMES ;DO 4000. ITERATIONS
5565
5577
(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) 024264 TEST17:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 024264 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 024272 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
(2) 024276 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 024304 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 024312 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 024320 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 024324 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 024332 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(1)
(2) ;*****
(2)
(2) ;SEIZE THE DRIVE THROUGH PORT A
(2)
(2) 024340 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 024346 013737 001216 001230 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 024354 005760 000000 TST RHCS1(RO) ;READ RHCS1
(3) 024360 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 024366 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 024374 013737 001220 001232 MOV PORTB,OPPR ;'OPPOSITE' PORT ADDRESS
(2) 024402 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
(2) 024410 010037 001122 MOV RO,SBADR ;RH11 BASE ADDRESS
(2) 024414 062737 000012 001122 ADD #RHDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
(2) 024422 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
(2) 024426 023737 001124 001126 CMP $GDDAT,SBDDAT ;IS THE REGISTER ZERO
(2) 024434 001403 BEQ .+10 ;BR IF IT IS
(2) 024436 104004 ERROR 4 ;REPORT THE ERROR
(2) 024440 000137 025070 JMP 1S ;BYPASS REST OF THE SUBTEST
(3) 024444 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 024452 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 024460 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 024466 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 024474 013737 001124 001160 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 024502 005137 001160 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 024506 013737 001126 001156 MOV SBDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 024514 043737 001160 001156 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(2) 024522 023737 001124 001156 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 024530 001401 BEQ .+4 ;BR IF THEY ARE
(2) 024532 104005 ERROR 5 ;REPORT THE ERROR
(1)
(2) ;*****

```

```

(2)
(2)
(2)
(3) 024534 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(3) 024542 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 024550 012760 000013 000000      MOV   #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3)
(3)
(3)
(3) 024556 005037 001242
(3) 024562 012737 000012 001122      CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 024570 060037 001122      MOV   #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 024574 012737 011700 001124      ADD   RO,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 024602 113760 001216 000010      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 024610 016037 000012 001162      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 024616 013737 001162 001156      MOV   RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 024624 042737 100100 001156      MOV   STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 024632 113760 001220 000010      BIC   #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 024640 016037 000012 001164      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 024646 013737 001164 001180      MOV   RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 024654 042737 100100 001160      MOV   STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 024662 023737 001156 001160      BIC   #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 024670 001006      CMP   STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 024672 005737 001156      BNE   64$ ;BR IF NOT
(3) 024676 001045      TST  STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 024700 104046      BNE  66$ ;BR IF NOT
(3) 024702 000137 025066      ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 024706 013737 001162 001126 64$: JMP  68$ ;BYPASS THE REST OF THE CHECKS
(3) 024714 013737 001220 001226      MOV   STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 024722 113760 001220 000010      MOV   PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 024730 005737 001156      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 024734 001414      TST  STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 024736 013737 001216 001226      BEQ  65$ ;BR IF ZERO
(3) 024744 013737 001164 001126      MOV   PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 024752 113760 001216 000010      MOV   STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 024760 005737 001160      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 024764 001012      TST  STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 024766 012737 177777 001242 65$: BNE  66$ ;BR IF NOT
(3) 024774 012760 000011 000000      MOV   #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 025002 012760 000013 000000      MOV   #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 025010 104026      MOV   #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 025012 013737 001162 001126 66$: ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 025020 013737 001216 001226      MOV   STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 025026 023737 001124 001162      MOV   PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 025034 001401      CMP   $GDDAT,STMP2 ;ALL BITS OK ?
(3) 025036 104007      BEQ  67$ ;BR IF OK FROM PORT A.
(3) 025040 013737 001164 001126 67$: ERROR 7 ;REPORT ERROR
(3) 025046 013737 001220 001226      MOV   STMP3,$BDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 025054 023737 001124 001164      MOV   PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 025062 001401      CMP   $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 025064 104007      BEQ  68$ ;BR IF OK
(3) 025066 000240      ERROR 7 ;REPORT ERROR
(2) 025070      NOP
(2)

```

```

(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 025070 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 025074 001412 BEQ TST20 ;BR IF NOT
(2) 025076 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 025104 001406 BEQ TST20 ;BR IF NOT
(2) 025106 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 025112 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 025116 000177 153766 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

5589  
5590

```

(3) ;*****
(3) *TEST 20 TEST SEIZE BY RHCSI READ THROUGH PORT 'B'
(4) *
(4) *VERIFY THAT READING THE CONTROL REGISTER (RHCSI) SEIZES THE DRIVE.
(4) *
(4) * A. READ THE CONTROL REGISTER (RHCSI) THROUGH PORT 'B'; VERIFY THAT
(4) * THE DRIVE IS SEIZED.
(4) *
(4) * B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
(4) * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
(4) *
(3) ;*****

```

```

(3) 025122 TST20: SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 025122 000004 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 025124 005737 001266 BEQ 25 ;BR IF NOT
(3) 025130 001406 BPL 15 ;BR IF JUST ENTERED TEST
(3) 025132 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 025134 000137 002410 15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 025140 012737 177777 001266 25: MOVB #20,$STNM ;TEST NUMBER
(3) 025146 112737 000020 001102 MOV #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 025154 012737 025176 001106 MOV #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 025162 012737 025176 001110 MOV #4000,$TIMES ;DO 4000. ITERATIONS
(1) 025170 012737 007640 001170

```

5591  
5592  
5593

```

(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)

```

```

(2) 025176 TEST20:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 025176 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 025204 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
(2) 025210 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 025216 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 025224 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 025232 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 025236 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 025244 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

```

```

(1) ;*****
(2)
(2)

```

```

(2)                                     ;SEIZE THE DRIVE THROUGH PORT B
(2)
(2) 025252 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 025260 013737 001220 001230      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 025266 005760 000000              TST    RHCSI(RO) ;READ RHCSI
(3) 025272 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(3) 025300 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 025306 013737 001216 001232      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 025314 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 025322 010037 001122              MOV    RO,SBADR ;RH11 BASE ADDRESS
(2) 025326 062737 000012 001122      ADD    #RHDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
(2) 025334 005037 001124              CLR    $GDDAT ;REGISTER SHOULD BE ZERO
(2) 025340 023737 001124 001126      CMP    $GDDAT,SBDDAT ;IS THE REGISTER ZERO
(2) 025346 001403                      BEQ    .+10 ;BR IF IT IS
(2) 025350 104004                      ERROR  4 ;REPORT THE ERROR
(2) 025352 000137 026002              JMP    IS ;BYPASS REST OF THE SUBTEST
(3) 025356 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 025364 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 025372 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 025400 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 025406 013737 001124 001160      MOV    $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 025414 005137 001160              COM    $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 025420 013737 001126 001156      MOV    SBDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 025426 043737 001160 001156      BIC    $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(2) 025434 023737 001124 001156      CMP    $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 025442 001401                      BEQ    .+4 ;BR IF THEY ARE
(2) 025444 104005                      ERROR  5 ;REPORT THE ERROR
(1)
(2)                                     ;*****
(2)
(2)                                     ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 025446 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 025454 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 025462 012760 000013 000000      MOV    #13,RHCSI(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 025470 005037 001242              CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 025474 012737 000012 001122      MOV    #RHDS1,SBADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 025502 060037 001122              ADD    RO,SBADR ;ADD THE I/O BASE ADDRESS
(3) 025506 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 025514 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 025522 016037 000012 001162      MOV    RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 025530 013737 001162 001156      MOV    $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
(3) 025536 042737 100100 001156      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 025544 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 025552 016037 000012 001164      MOV    RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 025560 013737 001164 001160      MOV    $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
(3) 025566 042737 100100 001160      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 025574 023737 001156 001160      CMP    $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 025602 001006                      BNE    64$ ;BR IF NOT
(3) 025604 005737 001156              TST    $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 025610 001045                      BNE    66$ ;BR IF NOT
    
```



```

(3) 025612 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 025614 000137 026000    JMP 68$          ;BYPASS THE REST OF THE CHECKS
(3) 025620 013737 001162 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 025626 013737 001220 001226    MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 025634 113760 001220 000010    MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 025642 005737 001156    TST $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 025646 001414    BEQ 65$        ;BR IF ZERO
(3) 025650 013737 001216 001226    MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 025656 013737 001164 001126    MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 025664 113760 001216 000010    MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 025672 005737 001160    TST $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 025676 001012    BNE 66$        ;BR IF NOT
(3) 025700 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 025706 012760 000011 000000    MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 025714 012760 000013 000000    MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 025722 104026          ERROR 26          ;TYPE ERROR MESSAGE 26
(3) 025724 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 025732 013737 001216 001226    MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 025740 023737 001124 001162    CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 025746 001401    BEQ 67$        ;BR IF OK FROM PORT A.
(3) 025750 104007          ERROR 7           ;REPORT ERROR
(3) 025752 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 025760 013737 001220 001226    MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 025766 023737 001124 001164    CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 025774 001401    BEQ 68$        ;BR IF OK
(3) 025776 104007          ERROR 7           ;REPORT ERROR
(3) 026000          000240    68$: NOP
(2) 026002          1$:
(2)
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 026002 105737 001103    TSTB $ERFLG     ;DID AN ERROR OCCUR ?
(4) 026006 001412    BEQ TST21       ;BR IF NOT
(2) 026010 032737 001000 177570    BIT #SW09,SWR   ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 026016 001406    BEQ TST21       ;BR IF NOT
(2) 026020 105037 001103    CLRB $ERFLG     ;CLEAR THE ERROR FLAG
(2) 026024 005037 001170    CLR $TIMES      ;CLEAR THE MAX ITERATION COUNT
(2) 026030 000177 153054    JMP $SLPERR     ;GO TO THE LOOP ADDRESS
    
```

5594  
5613  
5614

```

*****
;*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 RHDS1.
;*
;* B. WRITE 0'S INTO RHDS1 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'.
    
```

```

(4)
(4)
(4)
(4)
(3)
(3) 026034
(3) 026034 000004
(3) 026036 005737 001266
(3) 026042 001406
(3) 026044 100002
(3) 026046 000137 002410
(3) 026052 012737 177777 001266
(3) 026060 112737 000021 001102
(3) 026066 012737 026110 001106
(3) 026074 012737 026110 001110
(1) 026102 012737 007640 001170
5615
5648
(3)
(2)
(2) 026110
(2)
(2)
(2)
(2) 026110 113760 001216 000010
(2) 026116 005060 000012
(2) 026122 012760 000011 000000
(2) 026130 012760 000013 000000
(2) 026136 113760 001220 000010
(2) 026144 005060 000012
(2) 026150 012760 000011 000000
(2) 026156 012760 000013 000000
(1)
(2)
(2)
(2) 026164 113760 001220 000010
(2) 026172 013737 001220 001230
(2) 026200 005060 000012
(2) 026204 013737 001216 001232
(2) 026212 113760 001216 000010
(2) 026220 013737 001216 001226
(1)
(2)
(1)
(1)
(1) 026226 005060 000012
(1)
(2)
(1)
(1)
(2)

```

```

;*
;* D. ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
;*
*****
†ST21:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #21,$TSTNM ;TEST NUMBER
MOV #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;;DO 4000. ITERATIONS

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST21:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
CLR RHDS1(RO) ;SEIZE THE DRIVE
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RHDS1(RO) ;WRITE RHDS1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOVB PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

*****
;SET PORT REQUEST
CLR RHDS1(RO) ;SET PORT REQUEST FOR PORT A

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

```

```

(2)                                     ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 026232 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(3) 026240 013737 001220 001226      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026246 012760 000013 000000      MOV   #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 026254 005037 001242                CLR   RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 026260 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 026266 012737 000012 001122      MOV   #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 026274 060037 001122                ADD   RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 026300 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(4) 026306 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 026314 016037 000012 001156      MOV   RHDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
(4) 026322 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(4) 026330 013737 001220 001226      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 026336 016037 000012 001126      MOV   RHDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
(3) 026344 001404                BEQ   64$ ;BR IF STATUS FROM PORT B ZERO
(3) 026346 005737 001156                TST   $TMP0 ;IS STATUS FROM PORT A ZERO?
(3) 026352 001401                BEQ   64$ ;BR IF ZERO
(3) 026354 104031                ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 026356 013737 001156 001126 64$: MOV   $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
(3) 026364 013737 001216 001226      MOV   PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 026372 023737 001124 001126      CMP   $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 026400 001401                BEQ   .+4 ;BR IF OK
(3) 026402 104027                ERROR 27 ;REPORT REGISTER ERROR
(2) 026404 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(2) 026412 013737 001220 001226      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026420 005037 001236                CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 026424 016037 000012 001126      MOV   RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 026432 012737 000012 001122      MOV   #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 026440 060037 001122                ADD   RO,$BDADR ;ADD RH11 BASE ADDRESS
(2) 026444 005037 001124                CLR   $GDDAT ;WHAT REGISTER SHOULD BE
(2) 026450 013737 001126 001156      MOV   $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 026456 042737 077777 001156      BIC   #!CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 026464 023737 001124 001156      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 026472 001414                BEQ   65$ ;BR IF OK
(2) 026474 013737 001126 001166      MOV   $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 026502 042737 100000 001166      BIC   #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 026510 053737 001166 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 026516 104016                ERROR 16 ;TYPE MESSAGE 16
(2) 026520 005137 001236                COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 026524 000240                NOP
(2) 026526 113760 001216 000010 65$: MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(2) 026534 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026542 005037 001236                CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 026546 016037 000012 001126      MOV   RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 026554 012737 000012 001122      MOV   #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 026562 060037 001122                ADD   RO,$BDADR ;ADD RH11 BASE ADDRESS
(2) 026566 012737 100000 001124      MOV   #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 026574 013737 001126 001156      MOV   $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 026602 042737 077777 001156      BIC   #!CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 026610 023737 001124 001156      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS

```

```

(2) 026616 001414          BEQ      66$          ;BR IF OK
(2) 026620 013737 001126 001166    MOV      $BDDAT,STMP4 ;COPY 'BAD DATA'
(2) 026626 042737 100000 001166    BIC      #ATA,STMP4   ;CLEAR THE MASKED BITS
(2) 026634 053737 001166 001124    BIS      STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 026642 104016          ERROR    16          ;TYPE MESSAGE 16
(2) 026644 005137 001236          COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 026650 000240          66$:    NOP

(1)
(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT A
(3) 026652 113760 001216 000010    MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(3) 026660 013737 001216 001226    MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026666 012760 000013 000000    MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 026674 005037 001242          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 026700 012737 000012 001122    MOV      #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 026706 060037 001122          ADD     RO,$BDADR    ;ADD THE I/O BASE ADDRESS
(3) 026712 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 026720 113760 001216 000010    MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 026726 016037 000012 001162    MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 026734 013737 001162 001156    MOV      STMP2,STMP0   ;COPY IT INTO 'STMP0'
(3) 026742 042737 100100 001156    BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 026750 113760 001220 000010    MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 026756 016037 000012 001164    MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 026764 013737 001164 001160    MOV      STMP3,STMP1   ;COPY IT INTO 'STMP1'
(3) 026772 042737 100100 001160    BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 027000 023737 001156 001160    CMP      STMP0,STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 027006 001006          BNE     67$         ;BR IF NOT
(3) 027010 005737 001156          TST     STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 027014 001045          BNE     69$         ;BR IF NOT
(3) 027016 104046          ERROR    46         ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 027020 000137 027204          JMP     71$         ;BYPASS THE REST OF THE CHECKS
(3) 027024 013737 001162 001126 67$:  MOV      STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 027032 013737 001220 001226    MOV      PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 027040 113760 001220 000010    MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 027046 005737 001156          TST     STMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 027052 001414          BEQ     68$         ;BR IF ZERO
(3) 027054 013737 001216 001226    MOV      PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 027062 013737 001164 001126    MOV      STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 027070 113760 001216 000010    MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 027076 005737 001160          TST     STMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 027102 001012          BNE     69$         ;BR IF NOT
(3) 027104 012737 177777 001242 68$:  MOV      #-1,RELERR  ;SET 'RELEASE ERROR' INDICATOR
(3) 027112 012760 000011 000000    MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 027120 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 027126 104026          ERROR    26         ;TYPE ERROR MESSAGE 26
(3) 027130 013737 001162 001126 69$:  MOV      STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 027136 013737 001216 001226    MOV      PORTA,PTNBR  ;CHANGE PORT NUMBER
(3) 027144 023737 001124 001162    CMP      $GDDAT,STMP2 ;ALL BITS OK ?
(3) 027152 001401          BEQ     70$         ;BR IF OK FROM PORT A.

```

```

(3) 027154 104007          ERROR 7          ;REPORT ERROR
(3) 027156 013737 001164 001126 70$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 027164 013737 001220 001226      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 027172 023737 001124 001164      CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 027200 001401          BEQ 71$          ;BR IF OK
(3) 027202 104007          ERROR 7          ;REPORT ERROR
(3) 027204 000240          71$: NOP
(2) 027206          1$:

```

; IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 027206 105737 001103          TSTB $ERFLG      ;DID AN ERROR OCCUR ?
(4) 027212 001412          BEQ TST22        ;BR IF NOT
(2) 027214 032737 001000 177570    BIT $SW09,$SWR   ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 027222 001406          BEQ TST22        ;BR IF NOT
(2) 027224 105037 001103          CLRB $ERFLG     ;CLEAR THE ERROR FLAG
(2) 027230 005037 001170          CLR $TIMES      ;CLEAR THE MAX ITERATION COUNT
(2) 027234 000177 151650          JMP $SLPERR     ;GO TO THE LOOP ADDRESS

```

5667  
5668

```

*****
*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 RHDS1.
*
*  B. WRITE 0'S INTO RHDS1 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.
*
*****

```

```

(3) 027240          TST22:
(3) 027240 000004          SCOPE          ;INITIALIZE THE SCOPE HANDLER
(3) 027242 005737 001266    TST  KYBCTL     ;PERFORMING ONLY SINGLE TESTS ?
(3) 027246 001406          BEQ 2$          ;BR IF NOT
(3) 027250 100002          BPL 1$          ;BR IF JUST ENTERED TEST
(3) 027252 000137 002410    JMP EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 027256 012737 177777 001266    1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 027264 112737 000022 001102    2$: MOVB #22,$TSTNM ;TEST NUMBER
(3) 027272 012737 027314 001106    MOV #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 027300 012737 027314 001110    MOV #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 027306 012737 007640 001170    MOV #4000.,$TIMES ;DO 4000. ITERATIONS

```

5669  
5670

```

(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)

```

```

(2) 027314          TEST22:
(2)
(2)                ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 027314 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 027322 005060 000012              CLR    RHDS1(RO)       ;SEIZE THE DRIVE
(2) 027326 012760 000011 000000      MOV    #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 027334 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 027342 113760 001220 000010      MOVB   PORTB,RHCS2(RO);SELECT PORT #B
(2) 027350 005060 000012              CLR    RHDS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 027354 012760 000011 000000      MOV    #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 027362 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE
(1)
(2)                ;:*****
(2)
(2)                ;SEIZE THE DRIVE THROUGH PORT A
(2)
(2) 027370 113760 001216 000010      MOVB   PORTA,RHCS2(RO);SELECT PORT A
(2) 027376 013737 001216 001230      MOV    PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
(2) 027404 005060 000012              CLR    RHDS1(RO)     ;WRITE RHDS1
(2) 027410 013737 001220 001232      MOV    PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
(2) 027416 113760 001220 000010      MOVB   PORTB,RHCS2(RO);SELECT PORT B
(2) 027424 013737 001220 001226      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                ;:*****
(1)                ;SET PORT REQUEST
(1)
(1) 027432 005060 000012              CLR    RHDS1(RO)     ;SET PORT REQUEST FOR PORT B
(1)
(2)                ;:*****
(1)                ;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.
(1)
(2)
(2)                ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 027436 113760 001216 000010      MOVB   PORTA,RHCS2(RO);SELECT PORT A
(3) 027444 013737 001216 001226      MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027452 012760 000013 000000      MOV    #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3)                ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(3)
(3) 027460 005037 001242              CLR    RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 027464 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
(3) 027472 012737 000012 001122      MOV    #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 027500 060037 001122              ADD    RO,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 027504 113760 001220 000010      MOVB   PORTB,RHCS2(RO);SELECT PORT B
(4) 027512 013737 001220 001226      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 027520 016037 000012 001156      MOV    RHDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT B
(4) 027526 113760 001216 000010      MOVB   PORTA,RHCS2(RO);SELECT PORT A
(4) 027534 013737 001216 001226      MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 027542 016037 000012 001126      MOV    RHDS1(RO),$BDAT ;DRIVE STATUS FROM PORT A
(3) 027550 001404              BEQ    64$           ;BR IF STATUS FROM PORT A ZERO
(3) 027552 005737 001156              TST    $TMPD         ;IS STATUS FROM PORT B ZERO ?
(3) 027556 001401              BEQ    64$           ;BR IF ZERO
    
```

```

(3) 027560 104031          ERROR 31          ;REPORT DRIVE IN NEUTRAL
(3) 027562 013737 001156 001126 64$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
(3) 027570 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 027576 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 027604 001401 BEQ .+4 ;BR IF OK
(3) 027606 104027          ERROR 27          ;REPORT REGISTER ERROR
(2) 027610 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 027616 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027624 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 027630 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 027636 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 027644 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 027650 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 027654 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 027662 042737 077777 001156 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 027670 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 027676 001414 BEQ 65$ ;BR IF OK
(2) 027700 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 027706 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 027714 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 027722 104016          ERROR 16          ;TYPE MESSAGE 16
(2) 027724 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 027730 000240          NOP 65$:
(2) 027732 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 027740 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027746 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 027752 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 027760 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 027766 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 027772 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 030000 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 030006 042737 077777 001156 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 030014 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 030022 001414 BEQ 66$ ;BR IF OK
(2) 030024 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 030032 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 030040 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 030046 104016          ERROR 16          ;TYPE MESSAGE 16
(2) 030050 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 030054 000240          NOP 66$:
(1)
(2) ;*****
(2)
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 030056 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 030064 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 030072 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 030100 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 030104 012737 000012 001122 MOV #RHDS1,$BADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 030112 060037 001122 ADD RO,$BADR ;ADD THE I/O BASE ADDRESS

```

```

(3) 030116 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 030124 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT 'A'.
(3) 030132 016037 000012 001162      MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 030140 013737 001162 001156      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 030146 042737 100100 001156      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 030154 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 030162 016037 000012 001164      MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 030170 013737 001164 001160      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 030176 042737 100100 001160      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 030204 023737 001156 001160      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 030212 001006 001156 001160      BNE      67$ ;BR IF NOT
(3) 030214 005737 001156 001160      TST      STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 030220 001045 001156 001160      BNE      69$ ;BR IF NOT
(3) 030222 104046 001156 001160      ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 030224 000137 030410 001126 67$:      JMP      71$ ;BYPASS THE REST OF THE CHECKS
(3) 030230 013737 001162 001226 67$:      MOV      STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 030236 013737 001220 001226 67$:      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 030244 113760 001220 000010 67$:      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 030252 005737 001156 001160 67$:      TST      STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 030256 001414 001156 001160 67$:      BEQ      68$ ;BR IF ZERO
(3) 030260 013737 001216 001226 67$:      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 030266 013737 001164 001126 67$:      MOV      STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 030274 113760 001216 000010 67$:      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 030302 005737 001160 001160 67$:      TST      STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 030306 001012 001160 001160 67$:      BNE      69$ ;BR IF NOT
(3) 030310 012737 177777 001242 68$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 030316 012760 000011 000000 68$:      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 030324 012760 000013 000000 68$:      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 030332 104026 000013 000000 68$:      ERROR    26 ;TYPE ERROR MESSAGE 26
(3) 030334 013737 001162 001126 69$:      MOV      STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 030342 013737 001216 001226 69$:      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 030350 023737 001124 001162 69$:      CMP      $GDDAT,STMP2 ;ALL BITS OK ?
(3) 030356 001401 001124 001162 69$:      BEQ      70$ ;BR IF OK FROM PORT A.
(3) 030360 104007 001124 001162 69$:      ERROR    7 ;REPORT ERROR
(3) 030362 013737 001164 001126 70$:      MOV      STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 030370 013737 001220 001226 70$:      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 030376 023737 001124 001164 70$:      CMP      $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 030404 001401 001124 001164 70$:      BEQ      71$ ;BR IF OK
(3) 030406 104007 001124 001164 70$:      ERROR    7 ;REPORT ERROR
(3) 030410 000240 001124 001164 71$:      NOP
(2) 030412 000240 001124 001164 1$:
(2)
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 030412 105737 001103 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 030416 001412 001103 001103 BEQ TST23 ;BR IF NOT
(2) 030420 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 030426 001406 001000 177570 BEQ TST23 ;BR IF NOT
(2) 030430 105037 001103 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 030434 005037 001170 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 030440 000177 150444 150444 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

5671  
5687  
5688

\*\*\*\*\*



```

(3) ;*TEST 23 TEST NO 'PORT REQUEST' WHEN READ RHCS1 THROUGH PORT 'A'
(4) ;*
(4) ;*VERIFY THAT READING THE CONTROL REGISTER (RHCS1) DOES NOT SET 'PORT
(4) ;* REQUEST'.
(4) ;*
(4) ;* A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RHCS1. VERIFY THAT
(4) ;* THE DRIVE HAS BEEN SEIZED.
(4) ;*
(4) ;* B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT
(4) ;* SET.
(4) ;*
(4) ;* C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
(4) ;* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
(4) ;*
(3) ;*****

```

```

(3) ;TST23:
(3) 030444 000004 SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 030444 005737 001266 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 030446 001406 BEQ 2$ ;BR IF NOT
(3) 030452 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 030454 000137 002410 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 030456 012737 177777 001266 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 030462 012737 000023 001102 2$: MOVB #23,$TSTNM ;TEST NUMBER
(3) 030470 012737 030520 001106 MOV #TEST23,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 030476 012737 030520 001110 MOV #TEST23,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 030504 012737 030520 001110 MOV #4000.,$TIMES ;DO 4000. ITERATIONS
(1) 030512 012737 007640 001170

```

5689  
5709

```

(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2) ;*****

```

```

(2) 030520 TEST23:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 030520 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 030526 005060 000012 CLR RHCS1(RO) ;SEIZE THE DRIVE
(2) 030532 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 030540 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 030546 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 030554 005060 000012 CLR RHCS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 030560 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 030566 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

```

```

(1) ;*****
(2) ;*****

```

```

(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) 030574 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 030602 013737 001220 001230 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 030610 005760 000000 TST RHCS1(RO) ;READ RHCS1
(3) 030614 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 030622 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 030630 013737 001216 001232 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

```

```

(2) 030636 016037 000012 001126      MOV      RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 030644 010037 001122                MOV      RO,SBADR        ;RH11 BASE ADDRESS
(2) 030650 062737 000012 001122      ADD      #RHDS1,SBADR    ;GENERATE BAD REGISTER ADDRESS
(2) 030656 005037 001124                CLR      $GDDAT         ;REGISTER SHOULD BE ZERO
(2) 030662 023737 001124 001126      CMP      $GDDAT,SBDDAT  ;IS THE REGISTER ZERO
(2) 030670 001403                BEQ      .+10           ;BR IF IT IS
(2) 030672 104004                ERROR    4              ;REPORT THE ERROR
(2) 030674 000137 031446                JMP      IS             ;BYPASS REST OF THE SUBTEST
(3) 030700 113760 001220 000010      MOVVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 030706 013737 001220 001226      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 030714 016037 000012 001126      MOV      RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 030722 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 030730 013737 001124 001160      MOV      $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
(2) 030736 005137 001160                COM      $TMP1          ;COMPLEMENT THE EXPECTED STATUS
(2) 030742 013737 001126 001156      MOV      SBDDAT,$TMP0   ;SAVE THE ACTUAL STATUS
(2) 030750 043737 001160 001156      BIC      $TMP1,$TMP0    ;CLEAR UNWANTED BITS
(2) 030756 023737 001124 001156      CMP      $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
(2) 030764 001401                BEQ      .+4            ;BR IF THEY ARE
(2) 030766 104005                ERROR    5              ;REPORT THE ERROR
(2) 030770 113760 001216 000010      MOVVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 030776 013737 001216 001226      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)
(1)
(1)
(2) 031004 005037 001236                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 031010 016037 000000 001126      MOV      RHCS1(RO),SBDDAT ;GET CONTENTS OF RHCS1
(2) 031016 012737 000000 001122      MOV      #RHCS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 031024 060037 001122                ADD      RO,SBADR       ;ADD RH11 BASE ADDRESS
(2) 031030 005037 001124                CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
(2) 031034 013737 001126 001156      MOV      SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 031042 042737 173700 001156      BIC      #1C4077,$TMP0  ;SAVE SPECIFIED BITS
(2) 031050 023737 001124 001156      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 031056 001414                BEQ      64$           ;BR IF OK
(2) 031060 013737 001126 001166      MOV      SBDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 031066 042737 004077 001166      BIC      #4077,$TMP4    ;CLEAR THE MASKED BITS
(2) 031074 053737 001166 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 031102 104010                ERROR    10            ;REPORT THE ERROR
(2) 031104 005137 001236                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 031110 000240                64$: NOP
(1)
(2)
(1)
(1)
(2)
(2)
(2)
(2)
(3) 031112 113760 001220 000010      MOVVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 031120 013737 001220 001226      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 031126 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)
(3)
(3)
(3) 031134 005037 001242                CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR

```

\*\*\*\*\*  
;READ RHCS1 THROUGH PORT A - TRY TO SET PORT REQUEST

\*\*\*\*\*  
;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

(3) 031140 012737 000012 001122      MOV      #RHDS1,$BDADR      ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 031146 060037 001122                ADD      R0,$BDADR          ;ADD THE I/O BASE ADDRESS
(3) 031152 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 031160 113760 001216 000010      MOVVB   PORTA,RHCS2(R0)    ;SELECT PORT A.
(3) 031166 016037 000012 001162      MOV      RHDS1(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 031174 013737 001162 001156      MOV      $TMP2,$TMP0        ;COPY IT INTO '$TMP0'
(3) 031202 042737 100100 001156      BIC      #ATA!VV,$TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 031210 113760 001220 000010      MOVVB   PORTB,RHCS2(R0)    ;SELECT PORT B.
(3) 031216 016037 000012 001164      MOV      RHDS1(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 031224 013737 001164 001160      MOV      $TMP3,$TMP1        ;COPY IT INTO '$TMP1'
(3) 031232 042737 100100 001160      BIC      #ATA!VV,$TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 031240 023737 001156 001160      CMP      $TMP0,$TMP1        ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 031246 001006                BNE      65$                ;BR IF NOT
(3) 031250 005737 001156                TST      $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 031254 001045                BNE      67$                ;BR IF NOT
(3) 031256 104046                ERROR    46                 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 031260 000137 031444                JMP      69$                ;BYPASS THE REST OF THE CHECKS
(3) 031264 013737 001162 001126 65$:   MOV      $TMP2,$BDAT        ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 031272 013737 001220 001226      MOV      PORTB,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 031300 113760 001220 000010      MOVVB   PORTB,RHCS2(R0)    ;SELECT PORT B.
(3) 031306 005737 001156                TST      $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 031312 001414                BEQ      66$                ;BR IF ZERO
(3) 031314 013737 001216 001226      MOV      PORTA,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 031322 013737 001164 001126      MOV      $TMP3,$BDAT        ;'BAD DATA' FOR ERROR TYPE OUT
(3) 031330 113760 001216 000010      MOVVB   PORTA,RHCS2(R0)    ;SELECT PORT A.
(3) 031336 005737 001160                TST      $TMP1              ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 031342 001012                BNE      67$                ;BR IF NOT
(3) 031344 012737 177777 001242 66$:   MOV      #-1,RELEA         ;SET 'RELEASE ERROR' INDICATOR
(3) 031352 012760 000011 000000      MOV      #11,RHCS1(R0)     ;CLEAR THE DRIVE
(3) 031360 012760 000013 000000      MOV      #13,RHCS1(R0)     ;RELEASE THE DRIVE
(3) 031366 104026                ERROR    26                 ;TYPE ERROR MESSAGE 26
(3) 031370 013737 001162 001126 67$:   MOV      $TMP2,$BDAT        ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 031376 013737 001216 001226      MOV      PORTA,PTNBR        ;CHANGE PORT NUMBER
(3) 031404 023737 001124 001162      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
(3) 031412 001401                BEQ      68$                ;BR IF OK FROM PORT A.
(3) 031414 104007                ERROR    7                  ;REPORT ERROR
(3) 031416 013737 001164 001126 68$:   MOV      $TMP3,$BDAT        ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 031424 013737 001220 001226      MOV      PORTB,PTNBR        ;CHANGE PORT NUMBER
(3) 031432 023737 001124 001164      CMP      $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
(3) 031440 001401                BEQ      69$                ;BR IF OK
(3) 031442 104007                ERROR    7                  ;REPORT ERROR
(3) 031444 000240                69$:   NOP
(2) 031446                1$:
(2)
(2)
(2)
(2)
(2) 031446 105737 001103                TSTB    $ERFLG             ;DID AN ERROR OCCUR ?
(4) 031452 001412                BEQ      TST24              ;:BR IF NOT
(2) 031454 032737 001000 177570      BIT     #SW09,SWR           ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 031462 001406                BEQ      TST24              ;:BR IF NOT
(2) 031464 105037 001103                CLRB    $ERFLG             ;CLEAR THE ERROR FLAG
(2) 031470 005037 001170                CLR     $TIMES              ;CLEAR THE MAX ITERATION COUNT
(2) 031474 000177 147410                JMP     $SLPERR             ;GO TO THE LOOP ADDRESS

```



```

(2) 031664 013737 001220 001232 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 031672 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
(2) 031700 010037 001122 MOV RO,SBADR ;RH11 BASE ADDRESS
(2) 031704 062737 000012 001122 ADD #RHDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
(2) 031712 005037 001124 CLR SGDDAT ;REGISTER SHOULD BE ZERO
(2) 031716 023737 001124 001126 CMP SGDDAT,SBDDAT ;IS THE REGISTER ZERO
(2) 031724 001403 BEQ .+10 ;BR IF IT IS
(2) 031726 104004 ERROR 4 ;REPORT THE ERROR
(2) 031730 000137 032502 JMP 15 ;BYPASS REST OF THE SUBTEST
(3) 031734 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 031742 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 031750 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 031756 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,SGDDAT ;EXPECTED STATUS
(2) 031764 013737 001124 001160 MOV SGDDAT,STMP1 ;USE GOOD DATA AS A MASK
(2) 031772 005137 001160 COM STMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 031776 013737 001126 001156 MOV SBDDAT,STMP0 ;SAVE THE ACTUAL STATUS
(2) 032004 043737 001160 001156 BIC STMP1,STMP0 ;CLEAR UNWANTED BITS
(2) 032012 023737 001124 001156 CMP SGDDAT,STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 032020 001401 BEQ .+4 ;BR IF THEY ARE
(2) 032022 104005 ERROR 5 ;REPORT THE ERROR
(2) 032024 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 032032 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

(1)
(2)
(1) ;*****
(1) ;READ RHCS1 THROUGH PORT B - TRY TO SET PORT REQUEST

```

```

(2) 032040 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 032044 016037 000000 001126 MOV RHCS1(RO),SBDDAT ;GET CONTENTS OF RHCS1
(2) 032052 012737 000000 001122 MOV #RHCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 032060 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 032064 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
(2) 032070 013737 001126 001156 MOV SBDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 032076 042737 173700 001156 BIC #1C4077,STMP0 ;SAVE SPECIFIED BITS
(2) 032104 023737 001124 001156 CMP SGDDAT,STMP0 ;COMPARE THE BITS
(2) 032112 001414 BEQ 645 ;BR IF OK
(2) 032114 013737 001126 001166 MOV SBDDAT,STMP4 ;COPY 'BAD DATA'
(2) 032122 042737 004077 001166 BIC #4077,STMP4 ;CLEAR THE MASKED BITS
(2) 032130 053737 001166 001124 BIS STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 032136 104010 ERROR 10 ;REPORT THE ERROR
(2) 032140 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 032144 000240 645: NOP

```

```

(1)
(2) ;*****
(1) ;DRIVE SHOULD RETURN TO NEUTRAL

```

```

(2) ;RELEASE THE DRIVE FROM PORT A
(3) 032146 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 032154 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032162 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

|     |        |        |        |        |       |                   |                                                    |
|-----|--------|--------|--------|--------|-------|-------------------|----------------------------------------------------|
| (3) | 032170 | 005037 | 001242 |        | CLR   | RELERR            | :CLEAR THE 'RELEASE ERROR' INDICATOR               |
| (3) | 032174 | 012737 | 000012 | 001122 | MOV   | #RHDS1, \$BDADR   | :FORM THE ADDRESS OF RHDS1 FOR TYPEOUT             |
| (3) | 032202 | 060037 | 001122 |        | ADD   | RD, \$BDADR       | :ADD THE I/O BASE ADDRESS                          |
| (3) | 032206 | 012737 | 011700 | 001124 | MOV   | #MOL!PGM!DPR!DRY! | :VV, \$GDDAT ;COMPARISON CONSTANT                  |
| (3) | 032214 | 113760 | 001216 | 000010 | MOVB  | PORTA, RHCS2(RD)  | :SELECT PORT A.                                    |
| (3) | 032222 | 016037 | 000012 | 001162 | MOV   | RHDS1(RD), \$TMP2 | :GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| (3) | 032230 | 013737 | 001162 | 001156 | MOV   | \$TMP2, \$TMP0    | :COPY IT INTO 'TMP0'                               |
| (3) | 032236 | 042737 | 100100 | 001156 | BIC   | #ATA!VV, \$TMP0   | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| (3) | 032244 | 113760 | 001220 | 000010 | MOVB  | PORTB, RHCS2(RD)  | :SELECT PORT B.                                    |
| (3) | 032252 | 016037 | 000012 | 001164 | MOV   | RHDS1(RD), \$TMP3 | :GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| (3) | 032260 | 013737 | 001164 | 001160 | MOV   | \$TMP3, \$TMP1    | :COPY IT INTO 'TMP1'                               |
| (3) | 032266 | 042737 | 100100 | 001160 | BIC   | #ATA!VV, \$TMP1   | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| (3) | 032274 | 023737 | 001156 | 001160 | CMP   | \$TMP0, \$TMP1    | :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| (3) | 032302 | 001006 |        |        | BNE   | 65\$              | :BR IF NOT                                         |
| (3) | 032304 | 005737 | 001156 |        | TST   | \$TMP0            | :REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| (3) | 032310 | 001045 |        |        | BNE   | 67\$              | :BR IF NOT                                         |
| (3) | 032312 | 104046 |        |        | ERROR | 46                | :REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| (3) | 032314 | 000137 | 032500 |        | JMP   | 69\$              | :BYPASS THE REST OF THE CHECKS                     |
| (3) | 032320 | 013737 | 001162 | 001126 | MOV   | \$TMP2, \$BDADR   | :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| (3) | 032326 | 013737 | 001220 | 001226 | MOV   | PORTB, PTNBR      | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| (3) | 032334 | 113760 | 001220 | 000010 | MOVB  | PORTB, RHCS2(RD)  | :SELECT PORT B.                                    |
| (3) | 032342 | 005737 | 001156 |        | TST   | \$TMP0            | :SEE IF STATUS EQ 0 FROM PORT A.                   |
| (3) | 032346 | 001414 |        |        | BEQ   | 66\$              | :BR IF ZERO                                        |
| (3) | 032350 | 013737 | 001216 | 001226 | MOV   | PORTA, PTNBR      | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| (3) | 032356 | 013737 | 001164 | 001126 | MOV   | \$TMP3, \$BDADR   | : 'BAD DATA' FOR ERROR TYPE OUT                    |
| (3) | 032364 | 113760 | 001216 | 000010 | MOVB  | PORTA, RHCS2(RD)  | :SELECT PORT A.                                    |
| (3) | 032372 | 005737 | 001160 |        | TST   | \$TMP1            | :SEE IF STATUS EQ ZERO FROM PORT B.                |
| (3) | 032376 | 001012 |        |        | BNE   | 67\$              | :BR IF NOT                                         |
| (3) | 032400 | 012737 | 177777 | 001242 | MOV   | #-1, RELERR       | :SET 'RELEASE ERROR' INDICATOR                     |
| (3) | 032406 | 012760 | 000011 | 000000 | MOV   | #11, RHCS1(RD)    | :CLEAR THE DRIVE                                   |
| (3) | 032414 | 012760 | 000013 | 000000 | MOV   | #13, RHCS1(RD)    | :RELEASE THE DRIVE                                 |
| (3) | 032422 | 104026 |        |        | ERROR | 26                | :TYPE ERROR MESSAGE 26                             |
| (3) | 032424 | 013737 | 001162 | 001126 | MOV   | \$TMP2, \$BDADR   | :LOOK FOR BIT FAILURES WHEN RHDS1 READ             |
| (3) | 032432 | 013737 | 001216 | 001226 | MOV   | PORTA, PTNBR      | :CHANGE PORT NUMBER                                |
| (3) | 032440 | 023737 | 001124 | 001162 | CMP   | \$GDDAT, \$TMP2   | :ALL BITS OK ?                                     |
| (3) | 032446 | 001401 |        |        | BEQ   | 68\$              | :BR IF OK FROM PORT A.                             |
| (3) | 032450 | 104007 |        |        | ERROR | 7                 | :REPORT ERROR                                      |
| (3) | 032452 | 013737 | 001164 | 001126 | MOV   | \$TMP3, \$BDADR   | :CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.       |
| (3) | 032460 | 013737 | 001220 | 001226 | MOV   | PORTB, PTNBR      | :CHANGE PORT NUMBER                                |
| (3) | 032466 | 023737 | 001124 | 001164 | CMP   | \$GDDAT, \$TMP3   | :SEE IF READ OK FROM PORT B.                       |
| (3) | 032474 | 001401 |        |        | BEQ   | 69\$              | :BR IF OK                                          |
| (3) | 032476 | 104007 |        |        | ERROR | 7                 | :REPORT ERROR                                      |
| (3) | 032500 | 000240 |        |        | NOP   |                   |                                                    |
| (2) | 032502 |        |        |        |       |                   |                                                    |
| (2) |        |        |        |        |       |                   |                                                    |
| (2) |        |        |        |        |       |                   |                                                    |
| (2) |        |        |        |        |       |                   |                                                    |
| (2) | 032502 | 105737 | 001103 |        | TSTB  | SERFLG            | :DID AN ERROR OCCUR ?                              |
| (4) | 032506 | 001412 |        |        | BEQ   | TST25             | :BR IF NOT                                         |
| (2) | 032510 | 032737 | 001000 | 177570 | BIT   | #SW09, SWR        | :SEE IF LOOP ON ERROR SET (SWR9=1)                 |
| (4) | 032516 | 001406 |        |        | BEQ   | TST25             | :BR IF NOT                                         |
| (2) | 032520 | 105037 | 001103 |        | CLRB  | SERFLG            | :CLEAR THE ERROR FLAG                              |
| (2) | 032524 | 005037 | 001170 |        | CLR   | \$TIMES           | :CLEAR THE MAX ITERATION COUNT                     |
| (2) | 032530 | 000177 | 146354 |        | JMP   | \$SLPERR          | :GO TO THE LOOP ADDRESS                            |

; IF ERROR OCCURED, CHECK FOR LOOP ON TEST



```

(2) 032664 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 032672 013737 001220 001230      MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 032700 005060 000012              CLR    RHDS1(RO) ;WRITE RHDS1
(3) 032704 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(3) 032712 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032720 013737 001216 001232      MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 032726 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 032734 010037 001122              MOV    RO,SBADR ;RH11 BASE ADDRESS
(2) 032740 062737 000012 001122      ADD    #RHDS1,SBADR ;GENERATE BAD REGISTER ADDRESS
(2) 032746 005037 001124              CLR    $GDDAT ;REGISTER SHOULD BE ZERO
(2) 032752 023737 001124 001126      CMP    $GDDAT,SBDDAT ;IS THE REGISTER ZERO
(2) 032760 001403                      BEQ    .+10 ;BR IF IT IS
(2) 032762 104004                      ERROR  4 ;REPORT THE ERROR
(2) 032764 000137 033716              JMP    IS ;BYPASS REST OF THE SUBTEST
(3) 032770 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 032776 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 033004 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 033012 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 033020 013737 001124 001160      MOV    $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 033026 005137 001160              COM    $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 033032 013737 001126 001156      MOV    SBDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 033040 043737 001160 001156      BIC    $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(2) 033046 023737 001124 001156      CMP    $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 033054 001401                      BEQ    .+4 ;BR IF THEY ARE
(2) 033056 104005                      ERROR  5 ;REPORT THE ERROR
(1)
(2)
;*****
;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
(1)
(1)
(2) 033060 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 033066 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 033074 012760 000013 000000      MOV    #13,RHCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT A
(1)
(2)
;*****
;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
(1)
(1)
(2) 033102 005037 001236                      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 033106 016037 000012 001126      MOV    RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 033114 012737 000012 001122      MOV    #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 033122 060037 001122              ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 033126 005037 001124              CLR    $GDDAT ;WHAT REGISTER SHOULD BE
(2) 033132 023737 001124 001126      CMP    $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 033140 001403                      BEQ    64$ ;BR IF OK
(2) 033142 104010                      ERROR  10 ;REPORT THE ERROR
(2) 033144 005137 001236                      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 033150 016037 000000 001126 64$: MOV    RHCS1(RO),SBDDAT ;GET THE CONTENTS OF RHCS1
(2) 033156 032737 020000 001126      BIT    #MCPE,SBDDAT ;IS 'MCPE' SET ?
(2) 033164 001404                      BEQ    .+12 ;BR IF NOT
(2) 033166 104011                      ERROR  11 ;REPORT THE ERROR
(2) 033170 012760 040000 000000      MOV    #TRE,RHCS1(RO) ;CLEAR 'MCPE'
(1) 033176 005737 001236                      TST    CKERR ;WAS RHDS1 NON ZERO ?
(1) 033202 001402                      BEQ    .+6 ;CONTENTS OF RHDS1 SEEN BY PORT A
(1) 033204 000137 033716                      JMP    IS ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

```



```

(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 033210 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 033216 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 033224 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 033232 005037 001242 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 033236 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 033244 012737 000012 001122 MOV #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 033252 060037 001122 ADD RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 033256 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(4) 033264 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 033272 016037 000012 001156 MOV RHDS1(RO),$TMPD ;READ STATUS REGISTER FROM PORT A
(4) 033300 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(4) 033306 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 033314 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
(3) 033322 001404 BEQ 65$ ;BR IF STATUS FROM PORT B ZERO
(3) 033324 005737 001156 TST $TMPD ;IS STATUS FROM PORT A ZERO ?
(3) 033330 001401 BEQ 65$ ;BR IF ZERO
(3) 033332 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 033334 013737 001156 001126 65$: MOV $TMPD,$BDDAT ;CHECK STATUS FROM PORT A
(3) 033342 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 033350 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 033356 001401 BEQ +4 ;BR IF OK
(3) 033360 104027 ERROR 27 ;REPORT REGISTER ERROR
(2)
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 033362 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 033370 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 033376 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 033404 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 033410 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 033416 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 033422 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 033430 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 033436 016037 000012 001162 MOV RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 033444 013737 001162 001156 MOV $TMP2,$TMPD ;COPY IT INTO '$TMPD'
(3) 033452 042737 100100 001156 BIC #ATA!VV,$TMPD ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 033460 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 033466 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 033474 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 033502 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 033510 023737 001156 001160 CMP $TMPD,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 033516 001006 BNE 66$ ;BR IF NOT
(3) 033520 005737 001156 TST $TMPD ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 033524 001045 BNE 68$ ;BR IF NOT

```

```

(3) 033526 104046          ERROR 46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 033530 000137 033714    JMP 70$          ;BYPASS THE REST OF THE CHECKS
(3) 033534 013737 001162 0C1126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 033542 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 033550 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 033556 005737 001156 TST $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 033562 001414 BEQ 67$          ;BR IF ZERO
(3) 033564 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 033572 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 033600 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 033606 005737 001160 TST $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 033612 001012 BNE 68$          ;BR IF NOT
(3) 033614 012737 177777 001242 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 033622 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 033630 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 033636 104026 ERROR 26          ;TYPE ERROR MESSAGE 26
(3) 033640 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 033646 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 033654 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 033662 001401 BEQ 69$          ;BR IF OK FROM PORT A.
(3) 033664 104007 ERROR 7           ;REPORT ERROR
(3) 033666 013737 001164 001126 69$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 033674 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 033702 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 033710 001401 BEQ 70$          ;BR IF OK
(3) 033712 104007 ERROR 7           ;REPORT ERROR
(3) 033714 000240 70$: NOP
(2) 033716 15:

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 033716 105737 001103 TSTB $ERFLG      ;DID AN ERROR OCCUR ?
(4) 033722 001412 BEQ TST26        ;BR IF NOT
(2) 033724 032737 001000 177570 BIT #SW09,SWR    ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 033732 001406 BEQ TST26        ;BR IF NOT
(2) 033734 105037 001103 CLRB $ERFLG      ;CLEAR THE ERROR FLAG
(2) 033740 005037 001170 CLR $TIMES       ;CLEAR THE MAX ITERATION COUNT
(2) 033744 000177 145140 JMP $SLPERR      ;GO TO THE LOOP ADDRESS

```

5796  
5797

```

(3) ;*****
(4) ;*TEST 26 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
(4) ;*
(4) ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
(4) ;* IS SEIZED BY THE OTHER PORT.
(4) ;*
(4) ;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
(4) ;*
(4) ;* B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
(4) ;*
(4) ;* C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
(4) ;*
(4) ;* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED
(4) ;* TO PORT 'B'.
(4) ;*

```

(4) ;\* E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED  
(4) ;\* TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.  
(4) ;\*

(3) ;\*\*\*\*\*  
(3) TST26: ;\*\*\*\*\*

|     |        |        |        |        |         |                  |                                  |
|-----|--------|--------|--------|--------|---------|------------------|----------------------------------|
| (3) | 033750 | 000004 |        |        | SCOPE   |                  | ; INITIALIZE THE SCOPE HANDLER   |
| (3) | 033750 | 005737 | 001266 |        | TST     | KYBCTL           | ; PERFORMING ONLY SINGLE TESTS ? |
| (3) | 033752 | 001406 |        |        | BEQ     | 25               | ; BR IF NOT                      |
| (3) | 033756 | 100002 |        |        | BPL     | 15               | ; BR IF JUST ENTERED TEST        |
| (3) | 033760 | 000137 | 002410 |        | JMP     | EXEC             | ; RETURN & GET NEXT TEST NUMBER  |
| (3) | 033762 | 012737 | 177777 | 001266 | 15: MOV | #-1, KYBCTL      | ; SET SINGLE TEST INDICATOR      |
| (3) | 033766 | 112737 | 000026 | 001102 | 25: MOV | #26, \$TSTNM     | ; TEST NUMBER                    |
| (3) | 034002 | 012737 | 034024 | 001106 | MOV     | #TEST26, \$LPADR | ; LOAD LOOP ON TEST ADDRESS      |
| (3) | 034010 | 012737 | 034024 | 001110 | MOV     | #TEST26, \$LPERR | ; LOAD LOOP ON ERROR ADDRESS     |
| (1) | 034016 | 012737 | 007640 | 001170 | MOV     | #4000., \$TIMES  | ; DO 4000. ITERATIONS            |

5798  
5799  
(3) ;\*\*\*\*\*  
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST  
(2) ;\*\*\*\*\*

(2) 034024 TEST26: ;CLEAR ATTENTION BITS FOR BOTH PORTS

|     |        |        |        |        |     |                  |                                    |
|-----|--------|--------|--------|--------|-----|------------------|------------------------------------|
| (2) | 034024 | 113760 | 001216 | 000010 | MOV | PORTA, RHCS2(RO) | ; SELECT PORT #A                   |
| (2) | 034032 | 005060 | 000012 |        | CLR | RHDS1(RO)        | ; SEIZE THE DRIVE                  |
| (2) | 034036 | 012760 | 000011 | 000000 | MOV | #11, RHCS1(RO)   | ; ISSUE DRIVE CLEAR                |
| (2) | 034044 | 012760 | 000013 | 000000 | MOV | #13, RHCS1(RO)   | ; RELEASE THE DRIVE                |
| (2) | 034052 | 113760 | 001220 | 000010 | MOV | PORTB, RHCS2(RO) | ; SELECT PORT #B                   |
| (2) | 034060 | 005060 | 000012 |        | CLR | RHDS1(RO)        | ; SEIZE THE DRIVE THROUGH PORT 'B' |
| (2) | 034064 | 012760 | 000011 | 000000 | MOV | #11, RHCS1(RO)   | ; ISSUE DRIVE CLEAR                |
| (2) | 034072 | 012760 | 000013 | 000000 | MOV | #13, RHCS1(RO)   | ; RELEASE THE DRIVE                |

(1) ;\*\*\*\*\*  
(2) ;\*\*\*\*\*

(2) ;SEIZE THE DRIVE THROUGH PORT A

|     |        |        |        |        |       |                    |                                             |
|-----|--------|--------|--------|--------|-------|--------------------|---------------------------------------------|
| (2) | 034100 | 113760 | 001216 | 000010 | MOV   | PORTA, RHCS2(RO)   | ; SELECT PORT A                             |
| (2) | 034106 | 013737 | 001216 | 001230 | MOV   | PORTA, SEIZPT      | ; STORE SEIZING PORT'S ADDRESS              |
| (2) | 034114 | 005060 | 000012 |        | CLR   | RHDS1(RO)          | ; WRITE RHDS1                               |
| (3) | 034120 | 113760 | 001220 | 000010 | MOV   | PORTB, RHCS2(RO)   | ; SELECT PORT B                             |
| (3) | 034126 | 013737 | 001220 | 001226 | MOV   | PORTB, PTNBR       | ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| (2) | 034134 | 013737 | 001220 | 001232 | MOV   | PORTB, OPPRT       | ; 'OPPOSITE' PORT ADDRESS                   |
| (2) | 034142 | 016037 | 000012 | 001126 | MOV   | RHDS1(RO), \$BDDAT | ; SEE IF DRIVE SEIZED BY PORT A             |
| (2) | 034150 | 010037 | 001122 |        | MOV   | RD, \$BDADR        | ; R#11 BASE ADDRESS                         |
| (2) | 034154 | 062737 | 000012 | 001122 | ADD   | #RHDS1, \$BDADR    | ; GENERATE BAD REGISTER ADDRESS             |
| (2) | 034162 | 005037 | 001124 |        | CLR   | \$GDDAT            | ; REGISTER SHOULD BE ZERO                   |
| (2) | 034166 | 023737 | 001124 | 001126 | CMP   | \$GDDAT, \$BDDAT   | ; IS THE REGISTER ZERO                      |
| (2) | 034174 | 001403 |        |        | BEQ   | .+10               | ; BR IF IT IS                               |
| (2) | 034176 | 104004 |        |        | ERROR | 4                  | ; REPORT THE ERROR                          |
| (2) | 034200 | 000137 | 035132 |        | JMP   | 15                 | ; BYPASS REST OF THE SUBTEST                |
| (3) | 034204 | 113760 | 001216 | 000010 | MOV   | PORTA, RHCS2(RO)   | ; SELECT PORT A                             |
| (3) | 034212 | 013737 | 001216 | 001226 | MOV   | PORTA, PTNBR       | ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT |
| (2) | 034220 | 016037 | 000012 | 001126 | MOV   | RHDS1(RO), \$BDDAT | ; SEE IF SEIZING PORT SEES CORRECT STATUS   |

```

(2) 034226 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 034234 013737 001124 001160      MOV      $GDDAT,$STMP1 ;USE GOOD DATA AS A MASK
(2) 034242 005137 001160                COM      $STMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 034246 013737 001126 001156      MOV      $BDDAT,$STMP0 ;SAVE THE ACTUAL STATUS
(2) 034254 043737 001160 001156      BIC      $STMP1,$STMP0 ;CLEAR UNWANTED BITS
(2) 034262 023737 001124 001156      CMP      $GDDAT,$STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 034270 001401                BEQ      .+4 ;BR IF THEY ARE
(2) 034272 104005                ERROR 5 ;REPORT THE ERROR
(1)
(2) ;*****
(1) ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
(1)
(2) 034274 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(2) 034302 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 034310 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT B
(1)
(2) ;*****
(1) ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
(1)
(2) 034316 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 034322 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 034330 012737 000012 001122      MOV      #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 034336 060037 001122                ADD      RO,$BDADR ;ADD RH11 BASE ADDRESS
(2) 034342 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 034346 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 034354 001403                BEQ      64$ ;BR IF OK
(2) 034356 104010                ERROR 10 ;REPORT THE ERROR
(2) 034360 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 034364 016037 000000 001126 64$: MOV      RHCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
(2) 034372 032737 020000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET ?
(2) 034400 001404                BEQ      .+12 ;BR IF NOT
(2) 034402 104011                ERROR 11 ;REPORT THE ERROR
(2) 034404 012760 040000 000000      MOV      #TRE,RHCS1(RO) ;CLEAR 'MCPE'
(1) 034412 005737 001236                TST      CKERR ;WAS RHDS1 NON ZERO ?
(1) 034416 001402                BEQ      .+6 ;CONTENTS OF RHDS1 SEEN BY PORT B
(1) 034420 000137 035132                JMP      1$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
(1)
(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 034424 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(3) 034432 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 034440 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(3)
(3) 034446 005037 001242                CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 034452 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 034460 012737 000012 001122      MOV      #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 034466 060037 001122                ADD      RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 034472 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(4) 034500 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034506 016037 000012 001156      MOV      RHDS1(RO),$STMP0 ;READ STATUS REGISTER FROM PORT B

```

```

(4) 034514 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(4) 034522 013737 001216 001226      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034530 016037 000012 001126      MOV   RHDS1(RO),SBDDAT ;DRIVE STATUS FROM PORT A
(3) 034536 001404                BEQ   65$ ;BR IF STATUS FROM PORT A ZERO
(3) 034540 005737 001156                TST   $TMP0 ;IS STATUS FROM PORT B ZERO ?
(3) 034544 001401                BEQ   65$ ;BR IF ZERO
(3) 034546 104031                ERROR  31 ;REPORT DRIVE IN NEUTRAL
(3) 034550 013737 001156 001126 65$:  MOV   $TMP0,SBDDAT ;CHECK STATUS FROM PORT B
(3) 034556 013737 001220 001226      MOV   PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 034564 023737 001124 001126      CMP   $GDDAT,SBDDAT ;COMPARE WITH CONSTANT
(3) 034572 001401                BEQ   +4 ;BR IF OK
(3) 034574 104027                ERROR  27 ;REPORT REGISTER ERROR
(2)
(2)
(2) ;RELEASE THE DRIVE FROM PORT B
(3) 034576 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(3) 034604 013737 001220 001226      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 034612 012760 000013 000000      MOV   #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 034620 005037 001242                CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 034624 012737 000012 001122      MOV   #RHDS1,SBDDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 034632 060037 001122                ADD   RO,SBDDADR ;ADD THE I/O BASE ADDRESS
(3) 034636 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 034644 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 034652 016037 000012 001162      MOV   RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 034660 013737 001162 001156      MOV   $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 034666 042737 100100 001156      BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 034674 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 034702 016037 000012 001164      MOV   RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 034710 013737 001164 001160      MOV   $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 034716 042737 100100 001160      BIC   #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 034724 023737 001156 001160      CMP   $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 034732 001006                BNE   66$ ;BR IF NOT
(3) 034734 005737 001156                TST   $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 034740 001045                BNE   68$ ;BR IF NOT
(3) 034742 104046                ERROR  46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 034744 000137 035130                JMP   70$ ;BYPASS THE REST OF THE CHECKS
(3) 034750 013737 001162 001126 66$:  MOV   $TMP2,SBDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 034756 013737 001220 001226      MOV   PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 034764 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 034772 005737 001156                TST   $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 034776 001414                BEQ   67$ ;BR IF ZERO
(3) 035000 013737 001216 001226      MOV   PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 035006 013737 001164 001126      MOV   $TMP3,SBDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 035014 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 035022 005737 001160                TST   $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 035026 001012                BNE   68$ ;BR IF NOT
(3) 035030 012737 177777 001242 67$:  MOV   #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 035036 012760 000011 000000      MOV   #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 035044 012760 000013 000000      MOV   #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 035052 104026                ERROR  26 ;TYPE ERROR MESSAGE 26
(3) 035054 013737 001162 001126 68$:  MOV   $TMP2,SBDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ

```

```

(3) 035062 013737 001216 001226      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
(3) 035070 023737 001124 001162      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
(3) 035076 001401                      BEQ      69$              ;BR IF OK FROM PORT A.
(3) 035100 104007                      ERROR    7                ;REPORT ERROR
(3) 035102 013737 001164 001126 69$:  MOV      $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 035110 013737 001220 001226      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
(3) 035116 023737 001124 001164      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(3) 035124 001401                      BEQ      70$              ;BR IF OK
(3) 035126 104007                      ERROR    7                ;REPORT ERROR
(3) 035130 000240                      NOP
(2) 035132

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 035132 105737 001103      TSTB     $ERFLG          ;DID AN ERROR OCCUR ?
(4) 035136 001412                      BEQ      TST27            ;:BR IF NOT
(2) 035140 032737 001000 177570    BIT      $SW09,$SWR      ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 035146 001406                      BEQ      TST27            ;:BR IF NOT
(2) 035150 105037 001103      CLRB     $ERFLG          ;CLEAR THE ERROR FLAG
(2) 035154 005037 001170      CLR      $TIMES          ;CLEAR THE MAX ITERATION COUNT
(2) 035160 000177 143724      JMP      $SLPERR         ;GO TO THE LOOP ADDRESS

```

5800  
5815  
5816

```

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

```

```

(3) 035164 000004                      TST27: SCOPE              ;INITIALIZE THE SCOPE HANDLER
(3) 035164 005737 001266      TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
(3) 035166 001406                      BEQ      2$              ;BR IF NOT
(3) 035172 100002                      BPL      1$              ;BR IF JUST ENTERED TEST
(3) 035176 000137 002410      JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
(3) 035202 012737 177777 001266 1$:  MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
(3) 035210 112737 000027 001102 2$:  MOVB    #27,$TSTNM      ;TEST NUMBER
(3) 035216 012737 035240 001106      MOV      #TEST27,$LPADR  ;LOAD LOOP ON TEST ADDRESS
(3) 035224 012737 035240 001110      MOV      #TEST27,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
(1) 035232 012737 007640 001170      MOV      #4000.,$TIMES  ;;DO 4000. ITERATIONS

```

5817  
5872  
(3)  
(2)  
(2)

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
*****

```

```

(2) 035240          TEST27:
(2)
(2)                ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 035240 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 035246 005060 000012              CLR    RHDS1(RO)       ;SEIZE THE DRIVE
(2) 035252 012760 000011 000000      MOV    #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 035260 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 035266 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 035274 005060 000012              CLR    RHDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 035300 012760 000011 000000      MOV    #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 035306 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE
(1)
(2)                ;*****
(1)                ;SELECT DRIVE OTHER THAN THAT BEING TESTED
(1) 035314 113760 001222 000010      MOVB   PORTC,RHCS2(RO) ;SELECT DRIVE NOT BEING TESTED
(1) 035322 013737 001216 001230      MOV    PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
(1)
(2)                ;*****
(1)                ;WRITE THE DRIVE'S ATTENTION BIT
(1) 035330 013760 001224 000016      MOV    ASR1,RHAS(RO)  ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
(2) 035336 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 035344 013737 001216 001226      MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                ;*****
(1)                ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
(1) 035352 005760 000012              TST    RHDS1(RO)      ;SEE THE REGISTER THROUGH PORT A ?
(1) 035356 001014                      BNE    1$             ;BR IF YES
(2) 035360 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 035366 013737 001220 001226      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 035374 005760 000012              TST    RHDS1(RO)      ;SEE REGISTER THROUGH PORT B ?
(1) 035400 001021                      BNE    2$             ;BR IF YES
(1) 035402 104042                      ERROR  42             ;DRIVE NOT IN NEUTRAL OR SEIZED
(1) 035404 000137 037150              JMP    4$             ;BYPASS REST OF TEST
(2) 035410
(2) 035410 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 035416 013737 001220 001226      MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 035424 005760 000012              TST    RHDS1(RO)      ;REGISTER SHOULD BE ZERO THROUGH PORT B
(1) 035430 001002                      BNE    +6             ;BR IF STATUS REG IS NOT ZERO
(1) 035432 000137 036302              JMP    3$             ;STATUS REG IS ZERO
(1) 035436 104043                      ERROR  43             ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
(1) 035440 000137 037150              JMP    4$             ;BYPASS REST OF TEST
(1)
(2)                ;*****
(1)                ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
(1)
(2) 035444
(2) 035444 005037 001236 25:          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 035450 016037 000012 001126      MOV    RHDS1(RO),SBDAT ;GET CONTENTS OF RHDS1
(2) 035456 012737 000012 001122      MOV    #RHDS1,SBDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 035464 060037 001122              ADD    RO,SBDADR      ;ADD RH11 BASE ADDRESS

```

```

(2) 035470 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 035476 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 035504 042737 106077 001156 BIC #1C71700,$TMP0 ;SAVE SPECIFIED BITS
(2) 035512 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 035520 001414 BEQ 64$ ;BR IF OK
(2) 035522 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 035530 042737 071700 001166 BIC #71700,$TMP4 ;CLEAR THE MASKED BITS
(2) 035536 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 035544 104010 ERROR 10 ;REPORT THE ERROR
(2) 035546 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 035552 000240 NOP
(1) 035554 013737 001220 001230 MOV PORTB,SEIZPT ;ADDRESS FOR ERROR MESSAGE
(1) 035562 013737 001216 001232 MOV PORTA,OPPRT ;SAME AS ABOVE
(2)
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 035570 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 035576 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 035604 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 035612 005037 001242 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 035616 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 035624 012737 000012 001122 MOV #RHDS1,$BDAOR ;REGISTER ADDRESS INCREMENT
(3) 035632 060037 001122 ADD RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 035636 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(4) 035644 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 035652 016037 000012 001156 MOV RHDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
(4) 035660 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(4) 035666 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 035674 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
(3) 035702 001404 BEQ 65$ ;BR IF STATUS FROM PORT B ZERO
(3) 035704 005737 001156 TST $TMP0 ;IS STATUS FROM PORT A ZERO?
(3) 035710 001401 BEQ 65$ ;BR IF ZERO
(3) 035712 104044 ERROR 44 ;REPORT DRIVE NOT SEIZED BY PORT A
(3) 035714 013737 001156 001126 MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
(3) 035722 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 035730 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 035736 001401 BEQ +4 ;BR IF OK
(3) 035740 104027 ERROR 27 ;REPORT REGISTER ERROR
(2)
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 035742 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 035750 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 035756 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 035764 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 035770 012737 000012 001122 MOV #RHDS1,$BDAOR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 035776 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(3) 036002 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT

```

64\$:

65\$:



```

(3) 036010 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 036016 016037 000012 001162      MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 036024 013737 001162 001156      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 036032 042737 100100 001156      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036040 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 036046 016037 000012 001164      MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 036054 013737 001164 001160      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 036062 042737 100100 001160      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036070 023737 001156 001160      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 036076 001006 ;BR IF NOT
(3) 036100 005737 001156 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 036104 001045 ;BR IF NOT
(3) 036106 104046 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 036110 000137 036274 ;BYPASS THE REST OF THE CHECKS
(3) 036114 013737 001162 001126 66$: MOV      STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 036122 013737 001220 001226 ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 036130 113760 001220 000010      MOV      PORTB,PTNBR ;SELECT PORT B.
(3) 036136 005737 001156      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 036142 001414      TST      STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 036144 013737 001216 001226      BEQ      67$ ;BR IF ZERO
(3) 036152 013737 001164 001126      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 036160 113760 001216 000010      MOV      STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 036166 005737 001160      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 036172 001012      TST      STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 036174 012737 177777 001242 67$: BNE      68$ ;BR IF NOT
(3) 036202 012760 000011 000000      MOV      #1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
(3) 036210 012760 000013 000000      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 036216 104026 ;MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 036220 013737 001162 001126 68$: ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 036226 013737 001216 001226      MOV      STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 036234 023737 001124 001162      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 036242 001401      CMP      $GDDAT,STMP2 ;ALL BITS OK ?
(3) 036244 104007      BEQ      69$ ;BR IF OK FROM PORT A.
(3) 036246 013737 001164 001126 69$: ERROR 7 ;REPORT ERROR
(3) 036254 013737 001220 001226      MOV      STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 036262 023737 001124 001164      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 036270 001401      CMP      $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 036272 104007      BEQ      70$ ;BR IF OK
(3) 036274 000240 ;ERROR 7 ;REPORT ERROR
(1) 036276 000137 037150 70$: NOP
(1) JMP      4$

```

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

```

(2) 036302 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 036310 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 036316 005037 001236 ;CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 036322 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 036330 012737 000012 001122      MOV      #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 036336 060037 001122 ;ADD      RO,$BDDADR ;ADD RH11 BASE ADDRESS
(2) 036342 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 036350 013737 001126 001156      MOV      $BDDAT,STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 036356 042737 106077 001156      BIC      #1C71700,STMP0 ;SAVE SPECIFIED BITS

```



```

(3) 036704 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036712 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 036720 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 036726 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 036734 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036742 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 036750 001006 BNE 73$ ;BR IF NOT
(3) 036752 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 036756 001045 BNE 75$ ;BR IF NOT
(3) 036760 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 036762 000137 037146 JMP 77$ ;BYPASS THE REST OF THE CHECKS
(3) 036766 013737 001162 001126 73$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 036774 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 037002 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 037010 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 037014 001414 BEQ 74$ ;BR IF ZERO
(3) 037016 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 037024 013737 001164 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 037032 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 037040 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 037044 001012 BNE 75$ ;BR IF NOT
(3) 037046 012737 177777 001242 74$: MOV #1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 037054 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 037062 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 037070 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 037072 013737 001162 001126 75$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 037100 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 037106 023737 001124 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 037114 001401 BEQ 76$ ;BR IF OK FROM PORT A.
(3) 037116 104007 ERROR 7 ;REPORT ERROR
(3) 037120 013737 001164 001126 76$: MOV STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 037126 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 037134 023737 001124 001164 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 037142 001401 BEQ 77$ ;BR IF OK
(3) 037144 104007 ERROR 7 ;REPORT ERROR
(3) 037146 000240 77$: NOP
(2) 037150 4$:
(2)
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 037150 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 037154 001412 BEQ TST30 ;BR IF NOT
(2) 037156 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 037164 001406 BEQ TST30 ;BR IF NOT
(2) 037166 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 037172 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 037176 000177 141706 JMP $JLPERR ;GO TO THE LOOP ADDRESS
    
```

5873  
5885  
5886

```

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



# H10

```
(2) 037404 113760 001216 000010      MOVB   PORTA,RHCS2(RO)  ;SELECT PORT A.
(2) 037412 016037 000012 001162      MOV    RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 037420 013737 001162 001156      MOV    STMP2,STMP0     ;COPY IT INTO 'STMP0'
(2) 037426 042737 100100 001156      BIC   #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 037434 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 037442 016037 000012 001164      MOV    RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 037450 013737 001164 001160      MOV    STMP3,STMP1     ;COPY IT INTO 'STMP1'
(2) 037456 042737 100100 001160      BIC   #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 037464 023737 001156 001160      CMP   STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 037472 001006 001156 001156      BNE   64$             ;BR IF NOT
(2) 037474 005737 001156 001156      TST   STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 037500 001045 001156 001156      BNE   66$             ;BR IF NOT
(2) 037502 104046 001156 001156      ERROR 46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 037504 000137 037670 001156      JMP   68$             ;BYPASS THE REST OF THE CHECKS
(2) 037510 013737 001162 001126 64$:  MOV   STMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 037516 013737 001220 001226      MOV   PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 037524 113760 001220 000010      MOVB  PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 037532 005737 001156 001156      TST   STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 037536 001414 001156 001156      BEQ   65$             ;BR IF ZERO
(2) 037540 013737 001216 001226      MOV   PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 037546 013737 001164 001126      MOV   STMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(2) 037554 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 037562 005737 001160 001160      TST   STMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 037566 001012 001160 001160      BNE   66$             ;BR IF NOT
(2) 037570 012737 177777 001242 65$:  MOV   #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
(2) 037576 012760 000011 000000      MOV   #11,RHCS1(RO)   ;CLEAR THE DRIVE
(2) 037604 012760 000013 000000      MOV   #13,RHCS1(RO)   ;RELEASE THE DRIVE
(2) 037612 104021 000013 000000      ERROR 21              ;TYPE ERROR MESSAGE 21
(2) 037614 013737 001162 001126 66$:  MOV   STMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 037622 013737 001216 001226      MOV   PORTA,PTNBR     ;CHANGE PORT NUMBER
(2) 037630 023737 001124 001162      CMP   $GDDAT,STMP2   ;ALL BITS OK ?
(2) 037636 001401 001124 001162      BEQ   67$             ;BR IF OK FROM PORT A.
(2) 037640 104007 001124 001162      ERROR 7               ;REPORT ERROR
(2) 037642 013737 001164 001126 67$:  MOV   STMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 037650 013737 001220 001226      MOV   PORTB,PTNBR     ;CHANGE PORT NUMBER
(2) 037656 023737 001124 001164      CMP   $GDDAT,STMP3   ;SEE IF READ OK FROM PORT B.
(2) 037664 001401 001124 001164      BEQ   68$             ;BR IF OK
(2) 037666 104007 001124 001164      ERROR 7               ;REPORT ERROR
(2) 037670 000240 001124 001164 68$:  NOP

; IF ERROR OCCURED, CHECK FOR LOOP ON TEST

(2) 037672 105737 001103 001103      TSTB  $ERFLG          ;DID AN ERROR OCCUR ?
(4) 037676 001412 001103 001103      BEQ   TST31           ;:BR IF NOT
(2) 037700 032737 001000 177570      BIT   #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 037706 001406 001000 177570      BEQ   TST31           ;:BR IF NOT
(2) 037710 105037 001103 001103      CLRB  $ERFLG          ;CLEAR THE ERROR FLAG
(2) 037714 005037 001170 001170      CLR   $TIMES          ;CLEAR THE MAX ITERATION COUNT
(2) 037720 000177 141164 001170      JMP   $JLPERR         ;GO TO THE LOOP ADDRESS

5907
5921
5922
(3)
(4)
;*****
;*TEST 31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
;*
```

```

(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3) 037724
(3) 037724 000004
(3) 037726 005737 001266
(3) 037732 001406
(3) 037734 100002
(3) 037736 000137 002410
(3) 037742 012737 177777 001266 15:
(3) 037750 112737 000031 001102 25:
(3) 037756 012737 040000 001106
(3) 037764 012737 040000 001110
(1) 037772 012737 000004 001170
5923
5980
(3)
(2)
(2)
(2) 040000
(2)
(2)
(2)
(2) 040000 113760 001216 000010
(2) 040006 005060 000012
(2) 040012 012760 000011 000000
(2) 040020 012760 000013 000000
(2) 040026 113760 001220 000010
(2) 040034 005060 000012
(2) 040040 012760 000011 000000
(2) 040046 012760 000013 000000
(2)
(2)
(2)
(2) 040054 113760 001216 000010
(2) 040062 013737 001216 001230
(2) 040070 005060 000012
(2) 040074 013737 001220 001232
(1)
(2)
(1)
(1)
(1) 040102 012760 177777 000014
(2)
(3)

```

```

;VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
;*
;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
;*
;* B. WRITE 1'S INTO RHER1 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.
;*
;*****
†ST31:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #31,$STNM ;TEST NUMBER
MOV #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS

;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST31:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
CLR RHDS1(RO) ;SEIZE THE DRIVE
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
;*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RHDS1(RO) ;WRITE RHDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
;*****
;FORCE AN ERROR
MOV #-1,RHER1(RO) ;SET ERROR BITS
;*****

```

```

(2) ;START THE TIMER
(2)
(2) 040110 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 040114 012737 003720 001246 MOV #2000, WATCH ;SET WATCH TO 2000 MS
(2) 040122 113760 001220 000010 MOVB PORTB, RHCS2(RO) ;SELECT PORT B
(2) 040130 013737 001220 001226 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(1) ;WAIT FOR DRIVE TO TIMEOUT
(1)
(1) 040136 005760 000012 1$: TST RHDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
(1) 040142 001004 BNE 2$ ;BR IF DRIVE RELEASED
(1) 040144 005737 001246 TST WATCH ;WATCH AT ZERO ?
(1) 040150 001372 BNE 1$ ;BR IF NOT
(1) 040152 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
(2) 040154 2$: MOVB PORTA, RHCS2(RO) ;SELECT PORT A
(2) 040162 013737 001216 000010 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(1) ;THE ERROR BIT ('ERR') IN RHDS1 SHOULD STILL BE SET
(1)
(2) 040170 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 040174 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 040202 012737 000012 001122 MOV #RHDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 040210 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
(2) 040214 012737 040000 001124 MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 040222 013737 001126 001156 MOV $BDDAT, $TMPD ;MOVE REGISTER CONTENTS TO 'TMPD'
(2) 040230 042737 137777 001156 BIC #140000, $TMPD ;SAVE SPECIFIED BITS
(2) 040236 023737 001124 001156 CMP $GDDAT, $TMPD ;COMPARE THE BITS
(2) 040244 001414 BEQ 64$ ;BR IF OK
(2) 040246 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 040254 042737 040000 001166 BIC #40000, $TMP4 ;CLEAR THE MASKED BITS
(2) 040262 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 040270 104023 ERROR 23 ;TYPE MESSAGE 23
(2) 040272 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 040276 000240 64$: NOP
(1)
(2) ;*****
(1) ;THE ERROR REGISTER SHOULD CONTAIN 1'S
(1)
(2) 040300 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 040304 016037 000014 001126 MOV RHER1(RO), $BDDAT ;GET CONTENTS OF RHER1
(2) 040312 012737 000014 001122 MOV #RHER1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 040320 060037 001122 ADD RO, $BDADR ;ADD RH11 BASE ADDRESS
(2) 040324 012737 177777 001124 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 040332 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
(2) 040340 001403 BEQ 65$ ;BR IF OK
(2) 040342 104010 ERROR 10 ;REPORT THE ERROR
(2) 040344 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 040350 000240 65$: NOP
(1)
(2) ;*****
(1) ;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

```

```

(1)
(2) 040352 005037 001236          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 040356 016037 000012 001126  MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 040364 012737 000012 001122  MOV      #RHDS1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 040372 060037 001122          ADD      RO, $BDADR     ;ADD RH11 BASE ADDRESS
(2) 040376 012737 100000 001124  MOV      #ATA, $GDDAT   ;WHAT REGISTER SHOULD BE
(2) 040404 013737 001126 001156  MOV      $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 040412 042737 077777 001156  BIC      #+CATA, $TMP0  ;SAVE SPECIFIED BITS
(2) 040420 023737 001124 001156  CMP      $GDDAT, $TMP0  ;COMPARE THE BITS
(2) 040426 001414          BEQ      66$           ;BR IF OK
(2) 040430 013737 001126 001166  MOV      $BDDAT, $TMP4  ;COPY 'BAD DATA'
(2) 040436 042737 100000 001166  BIC      #ATA, $TMP4    ;CLEAR THE MASKED BITS
(2) 040444 053737 001166 001124  BIS      $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 040452 104041          ERROR   41           ;TYPE MESSAGE 41
(2) 040454 005137 001236          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 040460 000240          66$:  NOP

```

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

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

(2) 040462 005037 001242          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 040466 012737 000012 001122  MOV      #RHDS1, $BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 040474 060037 001122          ADD      RO, $BDADR     ;ADD THE I/O BASE ADDRESS
(2) 040500 012737 051700 001124  MOV      #51700, $GDDAT ;COMPARISON CONSTANT
(2) 040506 113760 001216 000010  MOV      PORTA, RHCS2(RO) ;SELECT PORT A.
(2) 040514 016037 000012 001162  MOV      RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 040522 013737 001162 001156  MOV      $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
(2) 040530 042737 100100 001156  BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 040536 113760 001220 000010  MOV      PORTB, RHCS2(RO) ;SELECT PORT B.
(2) 040544 016037 000012 001164  MOV      RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 040552 013737 001164 001160  MOV      $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
(2) 040560 042737 100100 001160  BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 040566 023737 001156 001160  CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 040574 001006          BNE      67$           ;BR IF NOT
(2) 040576 005737 001156          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 040602 001045          BNE      69$           ;BR IF NOT
(2) 040604 104046          ERROR   46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 040606 000137 041006          JMP      71$           ;BYPASS THE REST OF THE CHECKS
(2) 040612 013737 001162 001126 67$:  MOV      $TMP2, $BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 040620 013737 001220 001226  MOV      PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 040626 113760 001220 000010  MOV      PORTB, RHCS2(RO) ;SELECT PORT B.
(2) 040634 005737 001156          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 040640 001414          BEQ      68$           ;BR IF ZERO
(2) 040642 013737 001216 001226  MOV      PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 040650 013737 001164 001126  MOV      $TMP3, $BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
(2) 040656 113760 001216 000010  MOV      PORTA, RHCS2(RO) ;SELECT PORT A.
(2) 040664 005737 001160          TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 040670 001012          BNE      69$           ;BR IF NOT
(2) 040672 012737 177777 001242 68$:  MOV      #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
(2) 040700 012760 000011 000000  MOV      #11, RHCS1(RO) ;CLEAR THE DRIVE
(2) 040706 012760 000013 000000  MOV      #13, RHCS1(RO) ;RELEASE THE DRIVE
(2) 040714 104026          ERROR   26           ;TYPE ERROR MESSAGE 26

```



```

(2) 040716 013737 001162 001126 69S: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 040724 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 040732 042737 100000 001162 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(2) 040740 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 040746 001401 BEQ 70S ;BR IF OK FROM PORT A.
(2) 040750 104007 ERROR 7 ;REPORT ERROR
(2) 040752 013737 001164 001126 70S: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 040760 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 040766 042737 100000 001164 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(2) 040774 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 041002 001401 BEQ 71S ;BR IF OK
(2) 041004 104007 ERROR 7 ;REPORT ERROR
(2) 041006 000240 71S: NOP

```

```

(1)
(2) ;*****
(1) ;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET
(1)

```

```

(2) 041010 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 041016 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 041024 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 041030 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 041036 012737 000012 001122 MOV #RHDS1,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 041044 060037 001122 ADD RO,$BADR ;ADD RH11 BASE ADDRESS
(2) 041050 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 041054 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 041062 042737 077777 001156 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 041070 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 041076 001414 BEQ 72S ;BR IF OK
(2) 041100 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 041106 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 041114 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 041122 104052 ERROR 52 ;TYPE MESSAGE 52
(2) 041124 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 041130 000240 72S: NOP

```

```

(2) ;CLEAR ATTENTION BIT FOR PORT A
(2)

```

```

(2) 041132 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 041140 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
(2) 041144 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 041152 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 041160 3S:

```

```

(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)

```

```

(2) 041160 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
(4) 041164 001412 BEQ TST32 ;:BR IF NOT
(2) 041166 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 041174 001406 BEQ TST32 ;:BR IF NOT
(2) 041176 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
(2) 041202 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 041206 000177 137676 JMP @SLPERR ;GO TO THE LOOP ADDRESS

```

5994  
5995

;\*\*\*\*\*

```

(3) ;*TEST 32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
(4) ;*
(4) ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
(4) ;*
(4) ;* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
(4) ;*
(4) ;* B. WRITE 1'S INTO RHER1 THROUGH PORT 'B'.
(4) ;*
(4) ;* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO
(4) ;* NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
(4) ;* PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
(4) ;*
(3) ;*****

```

```

(3) TST32:
(3) 041212 000004 SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 041212 005737 001266 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 041214 001406 BEQ 2$ ;BR IF NOT
(3) 041222 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 041224 000137 002410 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 041230 012737 177777 001266 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 041236 112737 000032 001102 2$: MOVB #32,$STNM ;TEST NUMBER
(3) 041244 012737 041266 001106 MOV #TEST32,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 041252 012737 041266 001110 MOV #TEST32,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 041260 012737 000004 001170 MOV #4,$TIMES ;DO 4 ITERATIONS

```

```

5996 ;*****
5997 ;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

```

(2) TEST32:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 041266 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 041274 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
(2) 041300 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 041306 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 041314 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 041322 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 041326 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 041334 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

```

```

(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2)
(2) 041342 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 041350 013737 001220 001230 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 041356 005060 000012 CLR RHDS1(RO) ;WRITE RHDS1
(2) 041362 013737 001216 001232 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

```

```

(1) ;*****
(1) ;FORCE AN ERROR
(1)
(1) 041370 012760 177777 000014 MOV #-1,RHER1(RO) ;SET ERROR BITS

```

```

(2)
(3)
(2)
(2)
(2) 041376 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 041402 012737 003720 001246 MOV #2000,WATCH ;SET WATCH TO 2000 MS
(2) 041410 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(2) 041416 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)
(1)
(1)
(1) 041424 005760 000012 1$: TST RHDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
(1) 041430 001004 BNE 2$ ;BR IF DRIVE RELEASED
(1) 041432 005737 001246 TST WATCH ;WATCH AT ZERO ?
(1) 041436 001372 BNE 1$ ;BR IF NOT
(1) 041440 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
(2) 041442 2$:
(2) 041442 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 041450 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)
(1)
(1)
(2) 041456 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 041462 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 041470 012737 000012 001122 MOV #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 041476 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 041502 012737 040000 001124 MOV #ERR,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 041510 013737 001126 001156 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 041516 042737 137777 001156 BIC #1C4000,$TMP0 ;SAVE SPECIFIED BITS
(2) 041524 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 041532 001414 BEQ 64$ ;BR IF OK
(2) 041534 013737 001126 001166 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 041542 042737 040000 001166 BIC #4000,$TMP4 ;CLEAR THE MASKED BITS
(2) 041550 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 041556 104023 ERROR 23 ;TYPE MESSAGE 23
(2) 041560 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 041564 000240 64$: NOP
(1)
(2)
(1)
(1)
(2) 041566 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 041572 016037 000014 001126 MOV RHER1(RO),SBDDAT ;GET CONTENTS OF RHER1
(2) 041600 012737 000014 001122 MOV #RHER1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 041606 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 041612 012737 177777 001124 MOV #177777,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 041620 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 041626 001403 BEQ 65$ ;BR IF OK
(2) 041630 104010 ERROR 10 ;REPORT THE ERROR
(2) 041632 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 041636 000240 65$: NOP
(1)

```

```

(2)                                     ;*****
(1)                                     ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
(1)
(2) 041640 005037 001236                CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 041644 016037 000012 001126         MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 041652 012737 000012 001122         MOV      #RHDS1,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 041660 060037 001122                ADD      RO,SBADR      ;ADD RH11 BASE ADDRESS
(2) 041664 012737 100000 001124         MOV      #ATA,SGDDAT   ;WHAT REGISTER SHOULD BE
(2) 041672 013737 001126 001156         MOV      SBDDAT,STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 041700 042737 077777 001156         BIC      #ICATA,STMP0 ;SAVE SPECIFIED BITS
(2) 041706 023737 001124 001156         CMP      SGDDAT,STMP0 ;COMPARE THE BITS
(2) 041714 001414                        BEQ      66$           ;BR IF OK
(2) 041716 013737 001126 001166         MOV      SBDDAT,STMP4  ;COPY 'BAD DATA'
(2) 041724 042737 100000 001166         BIC      #ATA,STMP4   ;CLEAR THE MASKED BITS
(2) 041732 053737 001166 001124         BIS      STMP4,SGDDAT ;'OR' WITH GOOD DATA FOR TIMEOUT
(2) 041740 104041                        ERROR   41            ;TYPE MESSAGE 41
(2) 041742 005137 001236                COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 041746 000240                        66$:  NOP

(1)
(1)
(2)                                     ;*****
(2)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 041750 005037 001242                CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 041754 012737 000012 001122         MOV      #RHDS1,SBADR ;FORM THE ADDRESS OF RHDS1 FOR TIMEOUT
(2) 041762 060037 001122                ADD      RO,SBADR      ;ADD THE I/O BASE ADDRESS
(2) 041766 012737 051700 001124         MOV      #51700,SGDDAT ;COMPARISON CONSTANT
(2) 041774 113760 001216 000010         MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 042002 016037 000012 001162         MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 042010 013737 001162 001156         MOV      STMP2,STMP0   ;COPY IT INTO 'STMP0'
(2) 042016 042737 100100 001156         BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 042024 113760 001220 000010         MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 042032 016037 000012 001164         MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 042040 013737 001164 001160         MOV      STMP3,STMP1  ;COPY IT INTO 'STMP1'
(2) 042046 042737 100100 001160         BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 042054 023737 001156 001160         CMP      STMP0,STMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 042062 001006                        BNE      67$           ;BR IF NOT
(2) 042064 005737 001156                TST     STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 042070 001045                        BNE      69$           ;BR IF NOT
(2) 042072 104046                        ERROR   46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 042074 000137 042274                JMP      71$           ;BYPASS THE REST OF THE CHECKS
(2) 042100 013737 001162 001126 67$:  MOV      STMP2,SBDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 042106 013737 001220 001226         MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 042114 113760 001220 000010         MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 042122 005737 001156                TST     STMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 042126 001414                        BEQ      68$           ;BR IF ZERO
(2) 042130 013737 001216 001226         MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 042136 013737 001164 001126         MOV      STMP3,SBDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 042144 113760 001216 000010         MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 042152 005737 001160                TST     STMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 042156 001012                        BNE      69$           ;BR IF NOT
(2) 042160 012737 177777 001242 68$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
(2) 042166 012760 000011 000000         MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE

```

```

(2) 042174 012760 000013 000000      MOV      #13,RHCS1(RO)    ;RELEASE THE DRIVE
(2) 042202 104026                      ERROR      26             ;TYPE ERROR MESSAGE 26
(2) 042204 013737 001162 001126 69$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 042212 013737 001216 001226      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
(2) 042220 042737 100000 001162      BIC      #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
(2) 042226 023737 001124 001162      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
(2) 042234 001401                      BEQ      70$             ;BR IF OK FROM PORT A.
(2) 042236 104007                      ERROR      7             ;REPORT ERROR
(2) 042240 013737 001164 001126 70$:  MOV      $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 042246 013737 001220 001226      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
(2) 042254 042737 100000 001164      BIC      #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
(2) 042262 023737 001124 001164      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(2) 042270 001401                      BEQ      71$             ;BR IF OK
(2) 042272 104007                      ERROR      7             ;REPORT ERROR
(2) 042274 000240 71$:  NOP

(1)
(2)
(1)
(1)
(2) 042276 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 042304 013737 001216 001226      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 042312 005037 001236                      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 042316 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
(2) 042324 012737 000012 001122      MOV      #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 042332 060037 001122                      ADD      RO,$BDDADR      ;ADD RH11 BASE ADDRESS
(2) 042336 005037 001124                      CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
(2) 042342 013737 001126 001156      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 042350 042737 077777 001156      BIC      #ICATA,$TMP0    ;SAVE SPECIFIED BITS
(2) 042356 023737 001124 001156      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
(2) 042364 001414                      BEQ      72$             ;BR IF OK
(2) 042366 013737 001126 001166      MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
(2) 042374 042737 100000 001166      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
(2) 042402 053737 001166 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 042410 104052                      ERROR      52            ;TYPE MESSAGE 52
(2) 042412 005137 001236                      COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 042416 000240 72$:  NOP

(2)
(2)
(2)
(2)
(2) 042420 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 042426 005060 000012                      CLR      RHDS1(RO)       ;SEIZE THE DRIVE
(2) 042432 012760 000011 000000      MOV      #11,RHCS1(RO)   ;ISSUE DRIVE CLEAR
(2) 042440 012760 000013 000000      MOV      #13,RHCS1(RO)   ;RELEASE THE DRIVE
(2) 042446 3$:

(2)
(2)
(2)
(2)
(2) 042446 105737 001103                      TSTB     $ERFLG          ;DID AN ERROR OCCUR ?
(4) 042452 001412                      BEQ      TST33           ;:BR IF NOT
(2) 042454 032737 001000 177570      BIT      #SW09,$SWR      ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 042462 001406                      BEQ      TST33           ;:BR IF NOT
(2) 042464 105037 001103                      CLRB     $ERFLG          ;CLEAR THE ERROR FLAG
(2) 042470 005037 001170                      CLR      $TIMES          ;CLEAR THE MAX ITERATION COUNT
(2) 042474 000177 136410                      JMP      @SLPERR         ;GO TO THE LOOP ADDRESS
    
```

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

5998  
6018  
6019  
(3)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(3)  
(1)  
6020  
6058  
(3)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)

042500  
042500 000004  
042502 005737 001266  
042506 001406  
042510 100002  
042512 000137 002410  
042516 012737 177777 001266  
042524 112737 000033 001102  
042532 012737 042554 001106  
042540 012737 042554 001110  
042546 012737 007640 001170

```
*****
*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 RHDS1.
*
* B. WRITE 1'S INTO RHER1 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 RHER1 HAS NOT BEEN CLEARED.
*
* D. CLEAR RHER1 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.
*
*****
```

```
TST33:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #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
```

```
*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
*****
```

042554  
042554 113760 001216 000010  
042562 005060 000012  
042566 012760 000011 000000  
042574 012760 000013 000000  
042602 113760 001220 000010  
042610 005060 000012  
042614 012760 000011 000000  
042622 012760 000013 000000

```
TEST33:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
CLR RHDS1(RO) ;SEIZE THE DRIVE
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
```

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

```

(2) 042630 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 042636 013737 001216 001230      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 042644 005060 000012                CLR      RHDS1(RO) ;WRITE RHDS1
(2) 042650 013737 001220 001232      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(1)
(2)
(1)
(1)
(1)
(1) 042656 012760 177777 000014      MOV      #-1,RHER1(RO) ;SET ERROR BITS
(1) 042664 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE A RELEASE COMMAND
(2) 042672 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 042676 016037 000000 001126      MOV      RHCS1(RO),SBDDAT ;GET CONTENTS OF RHCS1
(2) 042704 012737 000000 001122      MOV      #RHCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 042712 060037 001122                ADD      RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 042716 012737 004012 001124      MOV      #4012,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 042724 013737 001126 001156      MOV      SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 042732 042737 173765 001156      BIC      #1C4012,$TMP0 ;SAVE SPECIFIED BITS
(2) 042740 023737 001124 001156      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 042746 001414                BEQ      64$ ;BR IF OK
(2) 042750 013737 001126 001166      MOV      SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 042756 042737 004012 001166      BIC      #4012,$TMP4 ;CLEAR THE MASKED BITS
(2) 042764 053737 001166 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 042772 104025                ERROR   25 ;TYPE MESSAGE 25
(2) 042774 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043000 000240                NOP
(1) 043002 005737 001236                TST      CKERR ;DID 'GO' BIT RESET ?
(1) 043006 001002                BNE      .+6 ;BR IF NOT
(1) 043010 000137 043050                JMP      1$ ;'GO' BIT RESET
(1) 043014 012760 000040 000010      MOV      #CLR,RHCS2(RO) ;INIT THE RH11
(2) 043022 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 043030 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 043036 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE THROUGH PORT A
(1) 043044 000137 043564                JMP      2$ ;BYPASS THE REST OF THE TEST
(1)
(2)
(1)
(1)
(2)
(2) 043050 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(2) 043056 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043064 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 043070 016037 000012 001126      MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 043076 012737 000012 001122      MOV      #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 043104 060037 001122                ADD      RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 043110 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 043114 023737 001124 001126      CMP      $GDDAT,$SBDDAT ;IS THE REGISTER OK ?
(2) 043122 001403                BEQ      65$ ;BR IF OK
(2) 043124 104024                ERROR   24 ;TYPE MESSAGE 24
(2) 043126 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043132 000240                NOP
(2) 043134 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 043142 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043150 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 043154 016037 000014 001126      MOV      RHER1(RO),SBDDAT ;GET CONTENTS OF RHER1

```

```

(2) 043162 012737 000014 001122      MOV      #RHER1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 043170 060037 001122      ADD      RO,$BDADR    ;ADD RH11 BASE ADDRESS
(2) 043174 012737 177777 001124      MOV      #177777,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 043202 023737 001124 001126      CMP      $GDDAT,$BDAT  ;IS THE REGISTER OK ?
(2) 043210 001403      BEQ      66$          ;BR IF OK
(2) 043212 104010      ERROR   10          ;REPORT THE ERROR
(2) 043214 005137 001236      COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043220 000240      NOP

(1)
(2) ;*****
(1) ;CLEAR THE ERRORS THROUGH PORT A
(1)
(1) 043222 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
(1)
(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 043230 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(3) 043236 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043244 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 043252 005037 001242      CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 043256 012737 000012 001122      MOV      #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 043264 060037 001122      ADD      RO,$BDADR    ;ADD THE I/O BASE ADDRESS
(3) 043270 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 043276 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 043304 016037 000012 001162      MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 043312 013737 001162 001156      MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 043320 042737 100100 001156      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 043326 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 043334 016037 000012 001164      MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 043342 013737 001164 001160      MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 043350 042737 100100 001160      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 043356 023737 001156 001160      CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 043364 001006      BNE     67$          ;BR IF NOT
(3) 043366 005737 001156      TST     STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 043372 001045      BNE     69$          ;BR IF NOT
(3) 043374 104046      ERROR   46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 043376 000137 043562      JMP     71$          ;BYPASS THE REST OF THE CHECKS
(3) 043402 013737 001162 001126 67$: MOV      STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 043410 013737 001220 001226      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 043416 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 043424 005737 001156      TST     STMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 043430 001414      BEQ     58$          ;BR IF ZERO
(3) 043432 013737 001216 001226      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 043440 013737 001164 001126      MOV      STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 043446 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 043454 005737 001160      TST     STMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 043460 001012      BNE     69$          ;BR IF NOT
(3) 043462 012737 177777 001242 68$: MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 043470 012760 000011 000000      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE

```





```

(3) 043656 012737 043672 001110
(1) 043664 012737 007640 001170
6080
6081
(2)
(2)
(2) 043672
(2)
(2)
(2) 043672 113760 001216 000010
(2) 043700 005060 000012
(2) 043704 012760 000011 000000
(2) 043712 012760 000013 000000
(2) 043720 113760 001220 000010
(2) 043726 005060 000012
(2) 043732 012760 000011 000000
(2) 043740 012760 000013 000000
(2)
(2)
(2)
(2) 043746 113760 001220 000010
(2) 043754 013737 001220 001230
(2) 043762 005060 000012
(2) 043766 013737 001216 001232
(1)
(2)
(1)
(1)
(1) 043774 012760 177777 000014
(1) 044002 012760 000013 000000
(2) 044010 005037 001236
(2) 044014 016037 000000 001126
(2) 044022 012737 000000 001122
(2) 044030 060037 001122
(2) 044034 012737 004012 001124
(2) 044042 013737 001126 001156
(2) 044050 042737 173765 001156
(2) 044056 023737 001124 001156
(2) 044064 001414
(2) 044066 013737 001126 001166
(2) 044074 042737 004012 001166
(2) 044102 053737 001166 001124
(2) 044110 104025
(2) 044112 005137 001236
(2) 044116 000240
(1) 044120 005737 001236
(1) 044124 001002
(1) 044126 000137 044166
(1) 044132 012760 000040 000010
(2) 044140 113760 001220 000010
(2) 044146 013737 001220 001226

```

```

MOV #TEST34,SLPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000.,$TIMES ;DO 4000. ITERATIONS

;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

TEST34:
;CLEAR ATTENTION BITS FOR BOTH PORTS

MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
CLR RHDS1(RO) ;SEIZE THE DRIVE
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
;*****

;SEIZE THE DRIVE THROUGH PORT B

MOVB PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RHDS1(RO) ;WRITE RHDS1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

;*****
;FORCE AN ERROR

MOV #-1,RHER1(RO) ;SET ERROR BITS
MOV #13,RHCS1(RO) ;ISSUE A RELEASE COMMAND
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHCS1(RO),SBDDAT ;GET CONTENTS OF RHCS1
MOV #RHCS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,SBADR ;ADD RH11 BASE ADDRESS
MOV #4012,$GDDAT ;WHAT REGISTER SHOULD BE
MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #1C4012,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 64$ ;BR IF OK
MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #4012,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 25 ;TYPE MESSAGE 25
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

64$:
NOP
TST CKERR ;DID 'GO' BIT RESET ?
BNE .+6 ;BR IF NOT
JMP 1$ ;'GO' BIT RESET
MOV #CLR,RHCS2(RO) ;INIT THE RH11
MOVB PORTB,RHCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

(1) 044154 012760 000013 000000      MOV    #13,RHCS1(RO) ;RELEASE THE DRIVE THROUGH PORT B
(1) 044162 000137 044702                JMP    25            ;BYPASS THE REST OF THE TEST
(1)
(2)                                     ;*****
(1)                                     ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
(1)
(2) 044166                                15:
(2) 044166 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 044174 013737 001216 001226      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 044202 005037 001236                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 044206 016037 000012 001126      MOV    RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
(2) 044214 012737 000012 001122      MOV    #RHDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 044222 060037 001122                ADD    RO,$B0ADR    ;ADD RH11 BASE ADDRESS
(2) 044226 005037 001124                CLR    $GDDAT      ;WHAT REGISTER SHOULD BE
(2) 044232 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 044240 001403                        BEQ    65$         ;BR IF OK
(2) 044242 104024                        ERROR  24          ;TYPE MESSAGE 24
(2) 044244 005137 001236                COM    CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 044250 000240                                65$:
(2) 044252 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 044260 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 044266 005037 001236                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 044272 016037 000014 001126      MOV    RHER1(RO), $BDDAT ;GET CONTENTS OF RHER1
(2) 044300 012737 000014 001122      MOV    #RHER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 044306 060037 001122                ADD    RO,$B0ADR    ;ADD RH11 BASE ADDRESS
(2) 044312 012737 177777 001124      MOV    #177777,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 044320 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 044326 001403                        BEQ    66$         ;BR IF OK
(2) 044330 104010                        ERROR  10          ;REPORT THE ERROR
(2) 044332 005137 001236                COM    CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 044336 000240                                66$:
(1)                                     ;*****
(2)                                     ;CLEAR THE ERRORS THROUGH PORT B
(1)
(1) 044340 012760 000011 000000      MOV    #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
(1)
(2)                                     ;*****
(2)                                     ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 044346 113760 001220 000010      MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 044354 013737 001220 001226      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 044362 012760 000013 000000      MOV    #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 044370 005037 001242                CLR    RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 044374 012737 000012 001122      MOV    #RHDS1,$B0ADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 044402 060037 001122                ADD    RO,$B0ADR    ;ADD THE I/O BASE ADDRESS
(3) 044406 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 044414 113760 001216 000010      MOVB   PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 044422 016037 000012 001162      MOV    RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 044430 013737 001162 001156      MOV    $TMP2,$TMP0  ;COPY IT INTO 'TMP0'

```

```

(3) 044436 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 044444 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 044452 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 044460 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 044466 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 044474 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 044502 001006 BNE 67$ ;BR IF NOT
(3) 044504 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 044510 001045 BNE 69$ ;BR IF NOT
(3) 044512 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 044514 000137 044700 JMP 71$ ;BYPASS THE REST OF THE CHECKS
(3) 044520 013737 001162 001126 67$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 044526 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 044534 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 044542 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 044546 001414 BEQ 68$ ;BR IF ZERO
(3) 044550 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 044556 013737 001164 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 044564 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 044572 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 044576 001012 BNE 69$ ;BR IF NOT
(3) 044600 012737 177777 001242 68$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 044606 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 044614 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 044622 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 044624 013737 001162 001126 69$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 044632 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 044640 023737 001124 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 044646 001401 BEQ 70$ ;BR IF OK FROM PORT A.
(3) 044650 104007 ERROR 7 ;REPORT ERROR
(3) 044652 013737 001164 001126 70$: MOV STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 044660 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 044666 023737 001124 001164 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 044674 001401 BEQ 71$ ;BR IF OK
(3) 044676 104007 ERROR 7 ;REPORT ERROR
(3) 044700 000240 71$: NOP
(2) 044702 2$:

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 044702 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 044706 001412 BEQ TST35 ;BR IF NOT
(2) 044710 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 044716 001406 BEQ TST35 ;BR IF NOT
(2) 044720 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 044724 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 044730 000177 134154 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

6092  
6098  
6099

```

;*****
;*TEST 35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
;*
;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
;*

```

- (4) :\* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
- (4) :\*
- (4) :\* B. WAIT 500 MS AND WRITE 0'S INTO RHDS1 THROUGH PORT 'A'.
- (4) :\*
- (4) :\* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
- (4) :\*
- (4) :\* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- (4) :\*

\*\*\*\*\*

TST35:

```

(3) 044734          000004          001266          1$: SCOPE          ;INITIALIZE THE SCOPE HANDLER
(3) 044734          000004          001266          TST          KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
(3) 044736          005737          001266          BEQ          2$          ;BR IF NOT
(3) 044742          001406          001266          BPL          1$          ;BR IF JUST ENTERED TEST
(3) 044744          100002          001266          JMP          EXEC          ;RETURN & GET NEXT TEST NUMBER
(3) 044746          000137          002410          1$: MOV          #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
(3) 044752          012737          177777          2$: MOV          #35,$TSTNM          ;TEST NUMBER
(3) 044760          112737          000035          001102          MOV          #TEST35,$LPADR          ;LOAD LOOP ON TEST ADDRESS
(3) 044766          012737          045010          001106          MOV          #TEST35,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
(3) 044774          012737          045010          001110          MOV          #4,$TIMES          ;;DO 4 ITERATIONS
(1) 045002          012737          000004          001170

```

6100  
6141

\*\*\*\*\*

;END OF 'SCOPE' SETUP - START OF MAIN TEST

TEST35:

```

(2) 045010          ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) MOV          PORTA,RHCS2(RO) ;SELECT PORT #A
(2) CLR          RHDS1(RO)       ;SEIZE THE DRIVE
(2) MOV          #11,RHCS1(RO)   ;ISSUE DRIVE CLEAR
(2) MOV          #13,RHCS1(RO)   ;RELEASE THE DRIVE
(2) MOV          PORTB,RHCS2(RO) ;SELECT PORT #B
(2) CLR          RHDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) MOV          #11,RHCS1(RO)   ;ISSUE DRIVE CLEAR
(2) MOV          #13,RHCS1(RO)   ;RELEASE THE DRIVE

```

\*\*\*\*\*

;SEIZE THE DRIVE THROUGH PORT A

```

(2) 045064          113760          001216          000010          MOV          PORTA,RHCS2(RO) ;SELECT PORT A
(2) 045072          013737          001216          001230          MOV          PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 045100          005060          000012          CLR          RHDS1(RO)       ;WRITE RHDS1
(2) 045104          013737          001220          001232          MOV          PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS

```

\*\*\*\*\*

;WAIT 500 MS

\*\*\*\*\*

(1)  
(2)  
(1)  
(1)  
(2)  
(3)

```

(2) ;START THE TIMER
(2)
(2) 045112 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 045116 012737 000764 001246 MOV #500.,WATCH ;SET WATCH TO 500 MS
(1) 045124 005737 001246 1$: TST WATCH ;WATCH EQUAL TO ZERO
(1) 045130 001375 BNE 1$ ;BR IF NOT
(2)
(3) ;*****
(2) ;START THE TIMER
(2)
(2) 045132 005037 001244 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 045136 012737 003720 001246 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
(1)
(2) ;*****
(1) ;RETRIGGER THE TIMEOUT ONE-SHOT
(1)
(1) 045144 005760 000012 TST RHDS1(RO) ;RETRIGGER THE ONE-SHOT
(2) 045150 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(2) 045156 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 045164 005760 000012 2$: TST RHDS1(RO) ;WAIT FOR TIMEOUT
(1) 045170 001004 BNE 3$ ;BR IF TIMEOUT OCCURED
(1) 045172 005737 001246 TST WATCH ;WATCH EQUAL TO ZERO ?
(1) 045176 001372 BNE 2$ ;BR IF NOT
(1) 045200 104036 ERROR 3$ ;NO TIMEOUT WITHIN 2 SECONDS
(1) 045202 013737 001244 001264 3$: MOV TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
(1)
(2) ;*****
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 045210 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
(2) 045214 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 045222 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(2) 045226 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 045234 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 045242 016037 000012 001162 MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 045250 013737 001162 001156 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
(2) 045256 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 045264 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 045272 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 045300 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(2) 045306 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 045314 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 045322 001006 BNE 64$ ;BR IF NOT
(2) 045324 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 045330 001045 BNE 66$ ;BR IF NOT
(2) 045332 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 045334 000137 045520 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(2) 045340 013737 001162 001126 64$: MOV STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 045346 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 045354 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 045362 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 045366 001414 BEQ 65$ ;BR IF ZERO
(2) 045370 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

(2) 045376 013737 001164 001126      MOV      $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
(2) 045404 113760 001216 000010      MOVVB   PORTA,RHCS2(RO)   ;SELECT PORT A.
(2) 045412 005737 001160              TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 045416 001012 001160              BNE     66$              ;BR IF NOT
(2) 045420 012737 177777 001242 65$:  MOV     #-1,RELEERR       ;SET 'RELEASE ERROR' INDICATOR
(2) 045426 012760 000011 000000      MOV     #11,RHCS1(RO)    ;CLEAR THE DRIVE
(2) 045434 012760 000013 000000      MOV     #13,RHCS1(RO)    ;RELEASE THE DRIVE
(2) 045442 104022 001162 001126 66$:  ERROR   22              ;TYPE ERROR MESSAGE 22
(2) 045444 013737 001162 001126      MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 045452 013737 001216 001226      MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
(2) 045460 023737 001124 001162      CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
(2) 045466 001401 001162 001126      BEQ     67$              ;BR IF OK FROM PORT A.
(2) 045470 104007 001164 001126 67$:  ERROR   7              ;REPORT ERROR
(2) 045472 013737 001164 001126      MOV     $TMP3,$BDDAT     ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 045500 013737 001220 001226      MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
(2) 045506 023737 001124 001164      CMP     $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
(2) 045514 001401 001164 001164      BEQ     68$              ;BR IF OK
(2) 045516 104007 001164 001164      ERROR   7              ;REPORT ERROR
(2) 045520 000240 001164 001164      NOP
    
```

```

(1)
(2) ;*****
(1) ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
(1)
    
```

```

(1) 045522 023737 001264 001252      CMP     TIMES,TIMEAP     ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
(1) 045530 003004 001264 001254      BGT     4$              ;BR IF GREATER
(1) 045532 023737 001264 001254      CMP     TIMES,TIMEAM     ;MEASURED TIME LESS THAN -25% TOLERANCE
(1) 045540 002001 001264 001254      BGE     .+4             ;BR IF NOT
(1) 045542 104025 001264 001254 4$:  ERROR   25             ;REPORT THE ERROR
    
```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 045544 105737 001103              TSTB   $ERFLG           ;DID AN ERROR OCCUR ?
(4) 045550 001412 001103 177570      BEQ     TST36            ;BR IF NOT
(2) 045552 032737 001000 177570      BIT     #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 045560 001406 001103 177570      BEQ     TST36            ;BR IF NOT
(2) 045562 105037 001103              CLRB   $ERFLG           ;CLEAR THE ERROR FLAG
(2) 045566 005037 001170              CLR    $TIMES           ;CLEAR THE MAX ITERATION COUNT
(2) 045572 000177 133312              JMP    @SLPERR          ;GO TO THE LOOP ADDRESS
    
```

```

6157
6158 ;*****
(3) *TEST 36      TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
(4) *
(4) *VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
(4) *
(4) * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
(4) *
(4) * B. WAIT 500 MS AND WRITE 0'B INTO RHDS1 THROUGH PORT 'A'.
(4) *
(4) * C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
(4) * TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
(4) *
(4) * D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
(4) * BIT IS SET.
(4) *
    
```

```

(3)          :*****
(3) 045576   †TST36:
(3) 045576   SCOPE
(3) 045600   000004   TST   KYBCTL   ; INITIALIZE THE SCOPE HANDLER
(3) 045600   005737   001266   BEQ   25       ; PERFORMING ONLY SINGLE TESTS ?
(3) 045604   001406   BPL   1$       ; BR IF NOT
(3) 045606   100002   JMP   EXEC     ; BR IF JUST ENTERED TEST
(3) 045610   000137   002410   MOV   #-1,KYBCTL ; RETURN & GET NEXT TEST NUMBER
(3) 045614   012737   177777   001266 1$:   MOV   #36,$TSTNM ; SET SINGLE TEST INDICATOR
(3) 045622   112737   000036   001102 2$:   MOV   #TEST36,$LPADR ; TEST NUMBER
(3) 045630   012737   045652   001106   MOV   #TEST36,$LPERR ; LOAD LOOP ON TEST ADDRESS
(3) 045636   012737   045652   001110   MOV   #4,$TIMES   ; LOAD LOOP ON ERROR ADDRESS
(1) 045644   012737   000004   001170   MOV   #4,$TIMES   ; DO 4 ITERATIONS

```

```

6159
6160
(3)          :*****
(2)          ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2)
(2)

```

```

(2) 045652   TEST36:
(2)          ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 045652   113760   001216   000010   MOV   PORTA,RHCS2(RO) ; SELECT PORT #A
(2) 045660   005060   000012   CLR   RHDS1(RO)       ; SEIZE THE DRIVE
(2) 045664   012760   000011   000000   MOV   #11,RHCS1(RO)   ; ISSUE DRIVE CLEAR
(2) 045672   012760   000013   000000   MOV   #13,RHCS1(RO)   ; RELEASE THE DRIVE
(2) 045700   113760   001220   000010   MOV   PORTB,RHCS2(RO) ; SELECT PORT #B
(2) 045706   005060   000012   CLR   RHDS1(RO)       ; SEIZE THE DRIVE THROUGH PORT 'B'
(2) 045712   012760   000011   000000   MOV   #11,RHCS1(RO)   ; ISSUE DRIVE CLEAR
(2) 045720   012760   000013   000000   MOV   #13,RHCS1(RO)   ; RELEASE THE DRIVE

```

```

(1)
(2)          :*****
(2)          ;SEIZE THE DRIVE THROUGH PORT B
(2)

```

```

(2) 045726   113760   001220   000010   MOV   PORTB,RHCS2(RO) ; SELECT PORT B
(2) 045734   013737   001220   001230   MOV   PORTB,SEIZPT    ; STORE SEIZING PORT'S ADDRESS
(2) 045742   005060   000012   CLR   RHDS1(RO)       ; WRITE RHDS1
(2) 045746   013737   001216   001232   MOV   PORTA,OPPRT     ; 'OPPOSITE' PORT ADDRESS

```

```

(1)
(2)          :*****
(1)          ;WAIT 500 MS
(1)

```

```

(2)          :*****
(2)          ;START THE TIMER
(2)

```

```

(2) 045754   005037   001244   CLR   TIME           ; CLEAR THE ELAPSED TIME COUNTER
(2) 045760   012737   000764   001246 1$:   MOV   #500.,WATCH   ; SET WATCH TO 500 MS
(1) 045766   005737   001246   TST   WATCH          ; WATCH EQUAL TO ZERO
(1) 045772   001375   BNE   1$             ; BR IF NOT

```

```

(2)          :*****
(2)          ;START THE TIMER

```



```

(2) 045774 005037 001244          CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 046000 012737 003720 001246  MOV      #2000.,WATCH ;SET WATCH TO 2000 MS
(1)
(2) ;*****
(1) ;RETRIGGER THE TIMEOUT ONE-SHOT
(1)
(1) 046006 005760 000012          TST      RHDS1(RO)    ;RETRIGGER THE ONE-SHOT
(2) 046012 113760 001216 000010  MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 046020 013737 001216 001226  MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 046026 005760 000012          25:     TST      RHDS1(RO)    ;WAIT FOR TIMEOUT
(1) 046032 001004          BNE     35            ;BR IF TIMEOUT OCCURED
(1) 046034 005737 001246          TST      WATCH       ;WATCH EQUAL TO ZERO ?
(1) 046040 001372          BNE     25            ;BR IF NOT
(1) 046042 104036          ERROR   36            ;NO TIMEOUT WITHIN 2 SECONDS
(1) 046044 013737 001244 001264  35:     MOV      TIME,TIMES  ;SAVE THE ELAPSED TIME VALUE
(1)
(2) ;*****
(2) ;
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 046052 005037 001242          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 046056 012737 000012 001122  MOV      #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(2) 046064 060037 001122          ADD     RO,$BDADR    ;ADD THE I/O BASE ADDRESS
(2) 046070 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 046076 113760 001216 000010  MOVB     PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 046104 016037 000012 001162  MOV      RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 046112 013737 001162 001156  MOV      STMP2,STMP0   ;COPY IT INTO 'STMP0'
(2) 046120 042737 100100 001156  BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 046126 113760 001220 000010  MOVB     PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 046134 016037 000012 001164  MOV      RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 046142 013737 001164 001160  MOV      STMP3,STMP1   ;COPY IT INTO 'STMP1'
(2) 046150 042737 100100 001160  BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 046156 023737 001156 001160  CMP      STMP0,STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 046164 001006          BNE     64$          ;BR IF NOT
(2) 046166 005737 001156          TST      STMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 046172 001045          BNE     66$          ;BR IF NOT
(2) 046174 104046          ERROR   46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 046176 000137 046362          JMP     68$          ;BYPASS THE REST OF THE CHECKS
(2) 046202 013737 001162 001126  64$:   MOV      STMP2,$DDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 046210 013737 001220 001226  MOV      PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 046216 113760 001220 000010  MOVB     PORTB,RHCS2(RO) ;SELECT PORT B.
(2) 046224 005737 001156          TST      STMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 046230 001414          BEQ     65$          ;BR IF ZERO
(2) 046232 013737 001216 001226  MOV      PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 046240 013737 001164 001126  MOV      STMP3,$DDAT   ;'BAD DATA' FOR ERROR TYPE OUT
(2) 046246 113760 001216 000010  MOVB     PORTA,RHCS2(RO) ;SELECT PORT A.
(2) 046254 005737 001160          TST      STMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 046260 001012          BNE     66$          ;BR IF NOT
(2) 046262 012737 177777 001242  65$:   MOV      #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
(2) 046270 012760 000011 000000  MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
(2) 046276 012760 000013 000000  MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 046304 104022          ERROR   22            ;TYPE ERROR MESSAGE 22
(2) 046306 013737 001162 001126  66$:   MOV      STMP2,$DDAT   ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(2) 046314 013737 001216 001226  MOV      PORTA,PTNBR  ;CHANGE PORT NUMBER

```

```

(2) 046322 023737 001124 001162      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
(2) 046330 001401                      BEQ      67$              ;BR IF OK FROM PORT A.
(2) 046332 104007                      ERROR   7                ;REPORT ERROR
(2) 046334 013737 001164 001126 67$:  MOV      $TMP3,$BDDAT     ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(2) 046342 013737 001220 001226      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
(2) 046350 023737 001124 001164      CMP      $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
(2) 046356 001401                      BEQ      68$              ;BR IF OK
(2) 046360 104007                      ERROR   7                ;REPORT ERROR
(2) 046362 000240 68$:  NOP

```

```

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

```

```

(1) 046364 023737 001264 001260      CMP      TIMES,TIMEBP     ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
(1) 046372 003004                      BGT      4$              ;BR IF GREATER
(1) 046374 023737 001264 001262      CMP      TIMES,TIMEBM     ;MEASURED TIME LESS THAN -25% TOLERANCE
(1) 046402 002001                      BGE     +4                ;BR IF NOT
(1) 046404 104025 4$:  ERROR   25                ;REPORT THE ERROR

```

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

```

(2) 046406 105737 001103              TSTB    $ERFLG           ;DID AN ERROR OCCUR ?
(4) 046412 001412                      BEQ      TST37           ;BR IF NOT
(2) 046414 032737 001000 177570      BIT     #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 046422 001406                      BEQ      TST37           ;BR IF NOT
(2) 046424 105037 001103              CLRB    $ERFLG           ;CLEAR THE ERROR FLAG
(2) 046430 005037 001170              CLR     $TIMES           ;CLEAR THE MAX ITERATION COUNT
(2) 046434 000177 132450              JMP     @SLPERR          ;GO TO THE LOOP ADDRESS

```

6161  
6182  
6183

```

;*****
*TEST 37      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 RHDS1.
*  B. SET PORT REQUEST BY WRITING 0'S INTO RHDS1 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.
*
;*****

```

```

(3) 046440                      TST37:
(3) 046440 000004              SCOPE
(3) 046442 005737 001266      TST     KYBCTL           ;INITIALIZE THE SCOPE HANDLER
;PERFORMING ONLY SINGLE TESTS ?

```

```

(3) 046446 001406          BEQ      25          ;BR IF NOT
(3) 046450 100002          BPL      15          ;BR IF JUST ENTERED TEST
(3) 046452 000137 002410    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 046456 012737 177777 001266 15:  MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
(3) 046464 112737 000037 001102 25:  MOVB     #37,$STSTNM  ;TEST NUMBER
(3) 046472 012737 046514 001106    MOV      #TEST37,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 046500 012737 046514 001110    MOV      #TEST37,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 046506 012737 000004 001170    MOV      #4,$TIMES    ;;DO 4 ITERATIONS
6184
6233
(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) 046514 TEST37:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 046514 113760 001216 000010    MOVB     PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 046522 005060 000012          CLR      RHDS1(RO)     ;SEIZE THE DRIVE
(2) 046526 012760 000011 000000    MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 046534 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 046542 113760 001220 000010    MOVB     PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 046550 005060 000012          CLR      RHDS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 046554 012760 000011 000000    MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 046562 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(1)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) 046570 113760 001220 000010    MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
(2) 046576 013737 001220 001230    MOV      PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
(2) 046604 005060 000012          CLR      RHDS1(RO)     ;WRITE RHDS1
(2) 046610 013737 001216 001232    MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
(2) 046616 113760 001216 000010    MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 046624 013737 001216 001226    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(1) ;SET REQUEST THROUGH PORT A
(1)
(1) 046632 005060 000012          CLR      RHDS1(RO)     ;SET REQUEST FOR PORT A
(2) 046636 113760 001220 000010    MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
(2) 046644 013737 001220 001226    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(1) ;RELEASE THE DRIVE THRU
(1)
(1) 046652 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE DRIVE THROUGH PORT B
(1)
(2) ;*****
(1) ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
(1)
(1) 046660 013737 001252 001246    MOV      TIMEAP,WATCH  ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
(1)

```

```

(2) ;:*****
(1) ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
(1)
(2) 046666 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 046672 016037 000012 001126 MOV RHCS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 046700 012737 000012 001122 MOV #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 046706 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 046712 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
(2) 046716 023737 001124 001126 CMP SGDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 046724 001403 BEQ 645 ;BR IF OK
(2) 046726 104031 ERROR 31 ;TYPE MESSAGE 31
(2) 046730 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 046734 000240 645: NOP
(1) 046736 005737 001236 TST CKERR ;REGISTER OK ?
(1) 046742 001402 BEQ .+6 ;BR IF OK
(1) 046744 000137 047402 JMP IS ;BYPASS REST OF TEST IF NOT
(1) 046750 005737 001246 TST WATCH ;WATCH EQUAL ZERO ?
(1) 046754 001375 BNE .-4 ;BR IF NOT
(1)
(2) ;:*****
(1) ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
(1)
(1) 046756 013737 001216 001226 MOV PORTA,PTNBR ;PORT NUMBER FOR TYPEOUT
(2) 046764 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 046770 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 046776 012737 000012 001122 MOV #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 047004 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 047010 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
(2) 047014 023737 001124 001126 CMP SGDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 047022 001403 BEQ 655 ;BR IF OK
(2) 047024 104035 ERROR 35 ;TYPE MESSAGE 35
(2) 047026 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 047032 000240 655: NOP
(1) 047034 005737 001236 TST CKERR ;REGISTER OK ?
(1) 047040 001402 BEQ .+6 ;BR IF OK
(1) 047042 000137 047402 JMP IS ;BYPASS REST OF TEST IF NOT
(1)
(2) ;:*****
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(2) MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 047046 113760 001216 000010 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 047054 013737 001216 001226 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(2) 047062 012760 000013 000000
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 047070 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 047074 012737 000012 001122 MOV #RHDS1,SBADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 047102 060037 001122 ADD RO,SBADR ;ADD THE I/O BASE ADDRESS
(3) 047106 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,SGDDAT ;COMPARISON CONSTANT
(3) 047114 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 047122 016037 000012 001162 MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 047130 013737 001162 001156 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'

```

```

(3) 047136 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 047144 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 047152 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 047160 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 047166 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 047174 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 047202 001006 BNE 66$ ;BR IF NOT
(3) 047204 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 047210 001045 BNE 68$ ;BR IF NOT
(3) 047212 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 047214 000137 047400 JMP 70$ ;BYPASS THE REST OF THE CHECKS
(3) 047220 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 047226 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 047234 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 047242 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 047246 001414 BEQ 67$ ;BR IF ZERO
(3) 047250 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 047256 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 047264 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 047272 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 047276 001012 BNE 68$ ;BR IF NOT
(3) 047300 012737 177777 001242 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 047306 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 047314 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 047322 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 047324 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 047332 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 047340 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 047346 001401 BEQ 69$ ;BR IF OK FROM PORT A.
(3) 047350 104007 ERROR 7 ;REPORT ERROR
(3) 047352 013737 001164 001126 69$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 047360 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 047366 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 047374 001401 BEQ 70$ ;BR IF OK
(3) 047376 104007 ERROR 7 ;REPORT ERROR
(3) 047400 000240 70$: NOP

```

```

(1)
(2) 047402 1$:
(2)
(2)

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

(2) 047402 105737 001103 TSTB SERFLG ;DID AN ERROR OCCUR ?
(4) 047406 001412 BEQ TST40 ;BR IF NOT
(2) 047410 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 047416 001406 BEQ TST40 ;BR IF NOT
(2) 047420 105037 001103 CLRB SERFLG ;CLEAR THE ERROR FLAG
(2) 047424 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 047430 000177 131454 JMP $JLPERR ;GO TO THE LOOP ADDRESS

```

6254  
6255

```

(3) *****
; *TEST 40 TEST NO TIMEOUT THROUGH PORT 'B'
(4) *
(4) ; *VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
(4) ; * SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

```

```

(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3) 047434
(3) 047434 000004
(3) 047436 005737 001266
(3) 047442 001406
(3) 047444 100002
(3) 047446 000137 002410
(3) 047452 012737 177777 001266
(3) 047460 112737 000040 001102
(3) 047466 012737 047510 001106
(3) 047474 012737 047510 001110
(1) 047502 012737 000004 001170
6256
6257
(3)
(2)
(2)
(2) 047510
(2)
(2)
(2)
(2) 047510 113760 001216 000010
(2) 047516 005060 000012
(2) 047522 012760 000011 000000
(2) 047530 012760 000013 000000
(2) 047536 113760 001220 000010
(2) 047544 005060 000012
(2) 047550 012760 000011 000000
(2) 047556 012760 000013 000000
(1)
(2)
(2)
(2)
(2) 047564 113760 001216 000010
(2) 047572 013737 001216 001230
(2) 047600 005060 000012
(2) 047604 013737 001220 001232
(2) 047612 113760 001220 000010

```

- ```

;*
;* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
;*
;* B. SET PORT REQUEST BY WRITING 0'S INTO RHDS1 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.
;*

```

```

*****
TST40:
SCOPE                               ;INITIALIZE THE SCOPE HANDLER
TST      KYBCTL                      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      2$                          ;BR IF NOT
BPL      1$                          ;BR IF JUST ENTERED TEST
JMP      EXEC                        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
2$:      MOVB     #40,$STSTM         ;TEST NUMBER
        MOV      #TEST40,$LPADR     ;LOAD LOOP ON TEST ADDRESS
        MOV      #TEST40,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
        MOV      #4,$TIMES          ;;DO 4 ITERATIONS

```

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

```

TEST40:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB     PORTA,RHCS2(RO) ;SELECT PORT #A
CLR      RHDS1(RO)      ;SEIZE THE DRIVE
MOV      #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
MOV      #13,RHCS1(RO)  ;RELEASE THE DRIVE
MOVB     PORTB,RHCS2(RO) ;SELECT PORT #B
CLR      RHDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV      #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
MOV      #13,RHCS1(RO)  ;RELEASE THE DRIVE

```

```

*****

```

```

;SEIZE THE DRIVE THROUGH PORT A
MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
MOV      PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
CLR      RHDS1(RO)      ;WRITE RHDS1
MOV      PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
MOVB     PORTB,RHCS2(RO) ;SELECT PORT B

```

```

(2) 047620 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                                     ;*****
(1)                                     ;SET REQUEST THROUGH PORT B
(1)
(1) 047626 005060 000012                CLR      RHDS1(RO) ;SET REQUEST FOR PORT B
(2) 047632 113760 001216 000010        MOV      PORTA,RHCS2(RO) ;SELECT PORT A
(2) 047640 013737 001216 001226        MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                                     ;*****
(1)                                     ;RELEASE THE DRIVE THR
(1)
(1) 047646 012760 000013 000000        MOV      #13,RHCS1(RO) ;RELEASE DRIVE THROUGH PORT A
(1)
(2)                                     ;*****
(1)                                     ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
(1)
(1) 047654 013737 001260 001246        MOV      TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
(1)
(2)                                     ;*****
(1)                                     ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
(1)
(2) 047662 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 047666 016037 000012 001126        MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 047674 012737 000012 001122        MOV      #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERRJR MESSAGE
(2) 047702 060037 001122                ADD      RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 047706 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 047712 023737 001124 001126        CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 047720 001403                        BEQ      64$ ;BR IF OK
(2) 047722 104031                        ERROR    31 ;TYPE MESSAGE 31
(2) 047724 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 047730 000240                        NOP
(1) 047732 005737 001236                TST      CKERR ;REGISTER OK ?
(1) 047736 001402                        BEQ      .+6 ;BR IF OK
(1) 047740 000137 050376                JMP      1$ ;BYPASS REST OF TEST IF NOT
(1) 047744 005737 001246                TST      WATCH ;WATCH EQUAL ZERO ?
(1) 047750 001375                        BNE      .-4 ;BR IF NOT
(1)
(2)                                     ;*****
(1)                                     ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
(1)
(1) 047752 013737 001220 001226        MOV      PORTB,PTNBR ;PORT NUMBER FOR TYPEOUT
(2) 047760 005037 001236                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 047764 016037 000012 001126        MOV      RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 047772 012737 000012 001122        MOV      #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 050000 060037 001122                ADD      RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 050004 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 050010 023737 001124 001126        CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 050016 001403                        BEQ      65$ ;BR IF OK
(2) 050020 104035                        ERROR    35 ;TYPE MESSAGE 35
(2) 050022 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 050026 000240                        NOP
(1) 050030 005737 001236                TST      CKERR ;REGISTER OK ?
(1) 050034 001402                        BEQ      .+6 ;BR IF OK

```

```

(1) 050036 000137 050376          JMP 15          ;BYPASS REST OF TEST IF NOT
(1)
(2)                               ;*****
(2)                               ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 050042 113760 001220 000010    MOVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 050050 013737 001220 001226    MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 050056 012760 000013 000000    MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                               ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 050064 005037 001242          CLR RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 050070 012737 000012 001122    MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 050076 060037 001122          ADD RO,$BDADR  ;ADD THE I/O BASE ADDRESS
(3) 050102 012737 011700 001124    MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 050110 113760 001216 000010    MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 050116 016037 000012 001162    MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 050124 013737 001162 001156    MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 050132 042737 100100 001156    BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 050140 113760 001220 000010    MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 050146 016037 000012 001164    MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 050154 013737 001164 001160    MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 050162 042737 100100 001160    BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 050170 023737 001156 001160    CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 050176 001006          BNE 66$        ;BR IF NOT
(3) 050200 005737 001156          TST STMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 050204 001045          BNE 68$        ;BR IF NOT
(3) 050206 104046          ERROR 46      ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 050210 000137 050374          JMP 70$        ;BYPASS THE REST OF THE CHECKS
(3) 050214 013737 001162 001126 66$: MOV STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 050222 013737 001220 001226    MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 050230 113760 001220 000010    MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 050236 005737 001156          TST STMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 050242 001414          BEQ 67$        ;BR IF ZERO
(3) 050244 013737 001216 001226    MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 050252 013737 001164 001126    MOV STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 050260 113760 001216 000010    MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 050266 005737 001160          TST STMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 050272 001012          BNE 68$        ;BR IF NOT
(3) 050274 012737 177777 001242 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 050302 012760 000011 000000    MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 050310 012760 000013 000000    MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 050316 104026          ERROR 26      ;TYPE ERROR MESSAGE 26
(3) 050320 013737 001162 001126 68$: MOV STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 050326 013737 001216 001226    MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 050334 023737 001124 001162    CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 050342 001401          BEQ 69$        ;BR IF OK FROM PORT A.
(3) 050344 104007          ERROR 7        ;REPORT ERROR
(3) 050346 013737 001164 001126 69$: MOV STMP3,$BDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 050354 013737 001220 001226    MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 050362 023737 001124 001164    CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 050370 001401          BEQ 70$        ;BR IF OK
(3) 050372 104007          ERROR 7        ;REPORT ERROR
    
```



```

(3) 050374 000240          70$:  NOP
(1)
(2) 050376          1$:
(2)
(2)          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2) 050376 105737 001103      TSTB  $ERFLG      ;DID AN ERROR OCCUR ?
(4) 050402 001412          BEQ    TST41      ;:BR IF NOT
(2) 050404 032737 001000 177570 BIT    #SW09,SWR  ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 050412 001406          BEQ    TST41      ;:BR IF NOT
(2) 050414 105037 001103      CLRB  $ERFLG      ;CLEAR THE ERROR FLAG
(2) 050420 005037 001170      CLR   $TIMES      ;CLEAR THE MAX ITERATION COUNT
(2) 050424 000177 130460      JMP   $SLPERR     ;GO TO THE LOOP ADDRESS

```

6258  
6275  
6276

```

(3) *****
(4) *TEST 41      TEST PORT 'A' ATTENTION AFTER A COMMAND
(4) *
(4) *
(4) *TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
(4) *COMMAND.
(4) *
(4) *  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
(4) *
(4) *  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
(4) *      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
(4) *      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
(4) *
(4) *  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
(4) *      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
(4) *
(3) *****

```

```

(3) 050430
(3) 050430 000004          TST41: SCOPE          ;INITIALIZE THE SCOPE HANDLER
(3) 050432 005737 001266      TST   KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
(3) 050436 001406          BEQ    2$         ;BR IF NOT
(3) 050440 100002          BPL   1$         ;BR IF JUST ENTERED TEST
(3) 050442 000137 002410      JMP   EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 050446 012737 177777 001266 1$: MOV   #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
(3) 050454 112737 000041 001102 2$: MOVB  #41,$STNM  ;TEST NUMBER
(3) 050462 012737 050504 001106      MOV   #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 050470 012737 050504 001110      MOV   #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 050476 012737 000004 001170      MOV   #4,$TIMES   ;;DO 4 ITERATIONS

```

6277  
6310

```

(3) *****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) TEST41:
(2)
(2)          ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 050504 113760 001216 000010      MOVB  PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 050512 005060 000012          CLR   RHDS1(RO)    ;SEIZE THE DRIVE

```

```

(2) 050516 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 050524 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 050532 113760 001220 000010      MOVB    PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 050540 005060 000012 000000      CLR     RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 050544 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 050552 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 050560 113760 001216 000010      MOVB    PORTA,RHCS2(RO) ;SELECT PORT A
(2) 050566 013737 001216 001226      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 050574 013737 001216 001230      MOV     PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
(1)
(2) ;:*****
(1) ;DO A RECALIBRATE THROUGH PORT A
(1)
(1) 050602 012760 000007 000000      MOV      #7,RHCS1(RO) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
(1)
(2) ;:*****
(1) ;WAIT FOR DRIVE TO FINISH
(1)
(1) 050610 032760 000200 000012      BIT     #DRY,RHDS1(RO) ;WAIT FOR DRIVE TO FINISH
(1) 050616 001774                BEQ     .-6 ;BR IF NOT FINISHED
(1)
(2) ;:*****
(1) ;CONFIRM THAT ATTENTION IS SET FOR PORT A
(1)
(2) 050620 005037 001236                CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 050624 016037 000012 001126      MOV     RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 050632 012737 000012 001122      MOV     #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 050640 060037 001122                ADD     RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 050644 012737 100000 001124      MOV     #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 050652 013737 001126 001156      MOV     SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 050660 042737 077777 001156      BIC     #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 050666 023737 001124 001156      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 050674 001414                BEQ     64$ ;BR IF OK
(2) 050676 013737 001126 001166      MOV     SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 050704 042737 100000 001166      BIC     #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 050712 053737 001166 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 050720 104032                ERROR  32 ;TYPE MESSAGE 32
(2) 050722 005137 001236                COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 050726 000240                NOP
(2) 050730 113760 001220 000010      MOVB    PORTB,RHCS2(RO) ;SELECT PORT B
(2) 050736 013737 001220 001226      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;:*****
(1) ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
(1)
(2) 050744 005037 001236                CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 050750 016037 000012 001126      MOV     RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
(2) 050756 012737 000012 001122      MOV     #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 050764 060037 001122                ADD     RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 050770 005037 001124                CLR     $GDDAT ;WHAT REGISTER SHOULD BE
(2) 050774 013737 001126 001156      MOV     SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 051002 042737 077777 001156      BIC     #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 051010 023737 001124 001156      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 051016 001414                BEQ     65$ ;BR IF OK

```

```

(2) 051020 013737 001126 001166      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
(2) 051026 042737 100000 001166      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
(2) 051034 053737 001166 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 051042 104032                ERROR    32                ;TYPE MESSAGE 32
(2) 051044 005137 001236                COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 051050 000240                65$:    NOP
(1)
(2)
(2)
(2)
(2)
(3) 051052 113760 001216 000010      MOV      PORTA,RHCS2(RO)  ;SELECT PORT A
(3) 051060 013737 001216 001226      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 051066 012760 000013 000000      MOV      #13,RHCS1(RO)   ;ISSUE RELEASE THROUGH PORT A
(3)
(3)
(3)
(3)
(3) 051074 005037 001242                CLR      RELERR           ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 051100 012737 000012 001122      MOV      #RHDS1,$BDDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 051106 060037 001122                ADD      RO,$BDDADR       ;ADD THE I/O BASE ADDRESS
(3) 051112 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 051120 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 051126 016037 000012 001162      MOV      RHDS1(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 051134 013737 001162 001156      MOV      $TMP2,$TMP0      ;COPY IT INTO 'TMP0'
(3) 051142 042737 100100 001156      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 051150 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 051156 016037 000012 001164      MOV      RHDS1(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 051164 013737 001164 001160      MOV      $TMP3,$TMP1     ;COPY IT INTO 'TMP1'
(3) 051172 042737 100100 001160      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 051200 023737 001156 001160      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 051206 001006                BNE      66$              ;BR IF NOT
(3) 051210 005737 001156                TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 051214 001045                BNE      68$              ;BR IF NOT
(3) 051216 104046                ERROR    46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 051220 000137 051404                JMP      70$              ;BYPASS THE REST OF THE CHECKS
(3) 051224 013737 001162 001126 66$:    MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 051232 013737 001220 001226      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 051240 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 051246 005737 001156                TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 051252 001414                BEQ      67$              ;BR IF ZERO
(3) 051254 013737 001216 001226      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 051262 013737 001164 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
(3) 051270 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 051276 005737 001160                TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 051302 001012                BNE      68$              ;BR IF NOT
(3) 051304 012737 177777 001242 67$:    MOV      #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
(3) 051312 012760 000011 000000      MOV      #11,RHCS1(RO)   ;CLEAR THE DRIVE
(3) 051320 012760 000013 000000      MOV      #13,RHCS1(RO)   ;RELEASE THE DRIVE
(3) 051326 104026                ERROR    26              ;TYPE ERROR MESSAGE 26
(3) 051330 013737 001162 001126 68$:    MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 051336 013737 001216 001226      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
(3) 051344 023737 001124 001162      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
(3) 051352 001401                BEQ      69$              ;BR IF OK FROM PORT A.
(3) 051354 104007                ERROR    7                ;REPORT ERROR

```

```

(3) 051356 013737 001164 001126 69$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 051364 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 051372 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 051400 001401 BEQ 70$ ;BR IF OK
(3) 051402 104007 ERROR 7 ;REPORT ERROR
(3) 051404 000240 70$: NOP
(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2)
(2) 051406 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
(4) 051412 001412 BEQ TST42 ;BR IF NOT
(2) 051414 032737 001000 177570 BIT #SW09,$SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 051422 001406 BEQ TST42 ;BR IF NOT
(2) 051424 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
(2) 051430 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 051434 000177 127450 JMP $SLPERR ;GO TO THE LOOP ADDRESS

```

6326  
6327

```

*****
*TEST 42 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.
*
*****

```

```

(3) 051440 TST42: SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 051440 000004 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 051442 005737 001266 BEQ 2$ ;BR IF NOT
(3) 051446 001406 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 051450 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 051452 000137 002410 MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 051456 012737 177777 001266 1$: MOV #42,$TSTNM ;TEST NUMBER
(3) 051464 112737 000042 001102 2$: MOVB #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 051472 012737 051514 001106 MOV #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 051500 012737 051514 001110 MOV #4,$TIMES ;DO 4 ITERATIONS
(1) 051506 012737 000004 001170

```

6328  
6329

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

```

(2) 051514 TEST42:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2)
(2) 051514 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT #A

```

```

(2) 051522 005060 000012          CLR      RHDS1(RO)      ;SEIZE THE DRIVE
(2) 051526 012760 000011 000000    MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 051534 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 051542 113760 001220 000010    MOVB    PORTB,RHCS2(RO);SELECT PORT #B
(2) 051550 005060 000012          CLR      RHDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 051554 012760 000011 000000    MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 051562 012760 000013 000000    MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 051570 113760 001220 000010    MOVB    PORTB,RHCS2(RO);SELECT PORT B
(2) 051576 013737 001220 001226    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 051604 013737 001220 001230    MOV      PORTB,SEIZPT  ;'SEIZED' PORT ADDRESS
(1)
(2)
(1)
(1)
(1)
(1) 051612 012760 000007 000000      MOV      #7,RHCS1(RO)  ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B
(1)
(2)
(1)
(1)
(1)
(1) 051620 032760 000200 000012      BIT      #DRY,RHDS1(RO);WAIT FOR DRIVE TO FINISH
(1) 051626 001774          BEQ      .-6           ;BR IF NOT FINISHED
(1)
(2)
(1)
(1)
(2) 051630 005037 001236          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 051634 016037 000012 001126    MOV      RHDS1(RO),SBDAT;GET CONTENTS OF RHDS1
(2) 051642 012737 000012 001122    MOV      #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 051650 060037 001122          ADD      RO,SBDADR     ;ADD RH11 BASE ADDRESS
(2) 051654 012737 100000 001124    MOV      #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
(2) 051662 013737 001126 001156    MOV      SBDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 051670 042737 077777 001156    BIC      #+CATA,$TMP0  ;SAVE SPECIFIED BITS
(2) 051676 023737 001124 001156    CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
(2) 051704 001414          BEQ      64$          ;BR IF OK
(2) 051706 013737 001126 001166    MOV      SBDAT,$TMP4   ;COPY 'BAD DATA'
(2) 051714 042737 100000 001166    BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
(2) 051722 053737 001166 001124    BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 051730 104032          ERROR    32           ;TYPE MESSAGE 32
(2) 051732 005137 001236          COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 051736 000240          NOP
(2) 051740 113760 001216 000010    MOVB    PORTA,RHCS2(RO);SELECT PORT A
(2) 051746 013737 001216 001226    MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)
(1)
(1)
(2)
(1)
(1)
(1)
(2) 051754 005037 001236          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 051760 016037 000012 001126    MOV      RHDS1(RO),SBDAT;GET CONTENTS OF RHDS1
(2) 051766 012737 000012 001122    MOV      #RHDS1,SBDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 051774 060037 001122          ADD      RO,SBDADR     ;ADD RH11 BASE ADDRESS
(2) 052000 005037 001124          CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
(2) 052004 013737 001126 001156    MOV      SBDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 052012 042737 077777 001156    BIC      #+CATA,$TMP0  ;SAVE SPECIFIED BITS
(2) 052020 023737 001124 001156    CMP      $GDDAT,$TMP0  ;COMPARE THE BITS

```

\*\*\*\*\*

;DO A RECALIBRATE THROUGH PORT B

\*\*\*\*\*

;WAIT FOR DRIVE TO FINISH

\*\*\*\*\*

;CONFIRM THAT ATTENTION IS SET FOR PORT B

64\$:

\*\*\*\*\*

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

```

(2) 052026 001414          BEQ      65$          ;BR IF OK
(2) 052030 013737 001126 001166      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 052036 042737 100000 001166      BIC      $ATA,$TMP4   ;CLEAR THE MASKED BITS
(2) 052044 053737 001166 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 052052 104032          ERROR    32          ;TYPE MESSAGE 32
(2) 052054 005137 001236          COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 052060 000240          65$:      NOP

(1)
(2)
(2)
(2)
(2)
(3) 052062 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
(3) 052070 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 052076 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)
(3)
(3) 052104 005037 001242          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 052110 012737 000012 001122      MOV      $RHDS1,$BQADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 052116 060037 001122          ADD      RO,$BQADR   ;ADD THE I/O BASE ADDRESS
(3) 052122 012737 011700 001124      MOV      $MOL!PGM!DPR!DRY!V,$GDDAT ;COMPARISON CONSTANT
(3) 052130 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 052136 016037 000012 001162      MOV      RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 052144 013737 001162 001156      MOV      $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
(3) 052152 042737 100100 001156      BIC      $ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 052160 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 052166 016037 000012 001164      MOV      RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 052174 013737 001164 001160      MOV      $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
(3) 052202 042737 100100 001160      BIC      $ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 052210 023737 001156 001160      CMP      $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 052216 001006          BNE      66$        ;BR IF NOT
(3) 052220 005737 001156          TST      $TMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 052224 001045          BNE      68$        ;BR IF NOT
(3) 052226 104046          ERROR    46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 052230 000137 052414          JMP      70$        ;BYPASS THE REST OF THE CHECKS
(3) 052234 013737 001162 001126 66$:      MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 052242 013737 001220 001226      MOV      PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 052250 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 052256 005737 001156          TST      $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 052262 001414          BEQ      67$        ;BR IF ZERO
(3) 052264 013737 001216 001226      MOV      PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 052272 013737 001164 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 052300 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 052306 005737 001160          TST      $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 052312 001012          BNE      68$        ;BR IF NOT
(3) 052314 012737 177777 001242 67$:      MOV      #-1,RELERR  ;SET 'RELEASE ERROR' INDICATOR
(3) 052322 012760 000011 000000      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 052330 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 052336 104026          ERROR    26          ;TYPE ERROR MESSAGE 26
(3) 052340 013737 001162 001126 68$:      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 052346 013737 001216 001226      MOV      PORTA,PTNBR  ;CHANGE PORT NUMBER
(3) 052354 023737 001124 001162      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 052362 001401          BEQ      69$        ;BR IF OK FROM PORT A.

```



```

(1) 052516 012737 007640 001170      MOV      #4000.,$TIMES      ;;DO 4000. ITERATIONS
6357
6393
(3)                                     ;:*****
(2)                                     ;:END OF 'SCOPE' SETUP - START OF MAIN TEST
(2)
(2) 052524      TEST43:
(2)
(2)                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 052524 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 052532 005060 000012 000010      CLR      RHDS1(RO)      ;SEIZE THE DRIVE
(2) 052536 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 052544 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 052552 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 052560 005060 000012 000010      CLR      RHDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 052564 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 052572 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
(2)
(2)                                     ;SEIZE THE DRIVE THROUGH PORT A
(2) 052600 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 052606 013737 001216 001230      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 052614 005060 000012 000010      CLR      RHDS1(RO)      ;WRITE RHDS1
(2) 052620 013737 001220 001232      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 052626 012760 177777 000014      MOV      #-1,RHER1(RO) ;LOAD 1'S INTO RHER1 THROUGH PORT A
(2) 052634 012760 177777 000040      MOV      #-1,RHER2(RO) ;LOAD 1'S INTO RHER2 THROUGH PORT A
(2) 052642 012760 177777 000042      MOV      #-1,RHER3(RO) ;LOAD 1'S INTO RHER3 THROUGH PORT A
(2) 052650 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
(2) 052656 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 052664 004737 053550 000010      JSR      PC,TST43B ;CHECK THE REGISTERS THROUGH PORT B
(2) 052670 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 052676 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 052704 005060 000042 000010      CLR      RHER3(RO) ;CLEAR RHER3 ON PORT A
(2) 052710 005060 000040 000010      CLR      RHER2(RO) ;CLEAR RHER2 ON PORT A
(2) 052714 005060 000014 000010      CLR      RHER1(RO) ;CLEAR RHER1 ON PORT A
(1) 052720 013760 001224 000016      MOV      ASR1,RHAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT A
(2) 052726 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
(2) 052734 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 052742 012760 177777 000014      MOV      #-1,RHER1(RO) ;LOAD 1'S INTO RHER1 THROUGH PORT B
(2) 052750 012760 177777 000040      MOV      #-1,RHER2(RO) ;LOAD 1'S INTO RHER2 THROUGH PORT B
(2) 052756 012760 177777 000042      MOV      #-1,RHER3(RO) ;LOAD 1'S INTO RHER3 THROUGH PORT B
(2) 052764 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(2) 052772 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 053000 004737 053550 000010      JSR      PC,TST43B ;CHECK THE REGISTERS THROUGH PORT A
(2)
(2)                                     ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 053004 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
(3) 053012 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 053020 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3)
(3)                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
    
```



```

(3) 053026 005037 001242 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 053032 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV $GDDAT ;COMPARISON CONSTANT
(3) 053040 012737 000012 001122 MOV #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 053046 060037 001122 ADD RO,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 053052 113760 001220 000010 MOVVB PORTB,RHCS2(RO) ;SELECT PORT B
(4) 053060 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 053066 016037 000012 001156 MOV RHDS1(RO),STMP0 ;READ STATUS REGISTER FROM PORT B
(4) 053074 113760 001216 000010 MOVVB PORTA,RHCS2(RO) ;SELECT PORT A
(4) 053102 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 053110 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
(3) 053116 001404 BEQ 64$ ;BR IF STATUS FROM PORT A ZERO
(3) 053120 005737 001156 TST STMP0 ;IS STATUS FROM PORT B ZERO ?
(3) 053124 001401 BEQ 64$ ;BR IF ZERO
(3) 053126 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 053130 013737 001156 001126 64$: MOV STMP0,$BDDAT ;CHECK STATUS FROM PORT B
(3) 053136 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 053144 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 053152 001401 BEQ +4 ;BR IF OK
(3) 053154 104027 ERROR 27 ;REPORT REGISTER ERROR
(2)
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 053156 113760 001220 000010 MOVVB PORTB,RHCS2(RO) ;SELECT PORT B
(3) 053164 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 053172 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 053200 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
(3) 053204 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 053212 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 053216 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 053224 113760 001216 000010 MOVVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 053232 016037 000012 001162 MOV RHDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 053240 013737 001162 001156 MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 053246 042737 100100 001156 BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 053254 113760 001220 000010 MOVVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 053262 016037 000012 001164 MOV RHDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 053270 013737 001164 001160 MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 053276 042737 100100 001160 BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 053304 023737 001156 001160 CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 053312 001006 BNE 65$ ;BR IF NOT
(3) 053314 005737 001156 TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 053320 001045 BNE 67$ ;BR IF NOT
(3) 053322 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 053324 000137 053510 JMP 69$ ;BYPASS THE REST OF THE CHECKS
(3) 053330 013737 001162 001126 65$: MOV STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 053336 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 053344 113760 001220 000010 MOVVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 053352 005737 001156 TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 053356 001414 BEQ 66$ ;BR IF ZERO
(3) 053360 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 053366 013737 001164 001126 MOV STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 053374 113760 001216 000010 MOVVB PORTA,RHCS2(RO) ;SELECT PORT A.

```

```

(3) 053402 005737 001160 TST STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 053406 001012 BNE 67$ ;BR IF NOT
(3) 053410 012737 177777 001242 66$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
(3) 053416 012760 000011 000000 MOV #11,RHCSI(RO) ;CLEAR THE DRIVE
(3) 053424 012760 000013 000000 MOV #13,RHCSI(RO) ;RELEASE THE DRIVE
(3) 053432 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 053434 013737 001162 001126 67$: MOV STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 053442 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 053450 023737 001124 001162 CMP $GDDAT,STMP2 ;ALL BITS OK ?
(3) 053456 001401 BEQ 68$ ;BR IF OK FROM PORT A.
(3) 053460 104007 ERROR 7 ;REPORT ERROR
(3) 053462 013737 001164 001126 68$: MOV STMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 053470 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 053476 023737 001124 001164 CMP $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 053504 001401 BEQ 69$ ;BR IF OK
(3) 053506 104007 ERROR 7 ;REPORT ERROR
(3) 053510 000240 69$: NOP

(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2)
(2) 053512 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
(4) 053516 001547 BEQ TST44 ;:BR IF NOT
(2) 053520 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 053526 001543 BEQ TST44 ;:BR IF NOT
(2) 053530 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
(2) 053534 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 053540 000177 125344 JMP @SLPERR ;GO TO THE LOOP ADDRESS
6394 053544 000137 054036 JMP TST44 ;GO TO THE NEXT TEST

(1) ;CHECK THE REGISTERS ON THE SELECTED PORT
(1)
(1)
(1) 053550 TST43B:
(3) 053550 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 053554 016037 000014 001126 MOV RHER1(RO),$BDDAT ;GET CONTENTS OF RHER1
(3) 053562 012737 000014 001122 MOV #RHER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 053570 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(3) 053574 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(3) 053600 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 053606 001403 BEQ 64$ ;BR IF OK
(3) 053610 104006 ERROR 6 ;TYPE MESSAGE 6
(3) 053612 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 053616 016037 000000 001126 64$: MOV RHCSI(RO),$BDDAT ;GET THE CONTENTS OF RHCSI
(3) 053624 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 053632 001404 BEQ +12 ;BR IF NOT
(3) 053634 104011 ERROR 11 ;REPORT THE ERROR
(3) 053636 012760 040000 000000 MOV #TRE,RHCSI(RO) ;CLEAR 'MCPE'
(3) 053644 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 053650 016037 000040 001126 MOV RHER2(RO),$BDDAT ;GET CONTENTS OF RHER2
(3) 053656 012737 000040 001122 MOV #RHER2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 053664 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(3) 053670 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(3) 053674 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 053702 001403 BEQ 65$ ;BR IF OK
(3) 053704 104006 ERROR 6 ;TYPE MESSAGE 6
    
```

```

(3) 053706 005137 001236          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 053712 016037 000000 001126 65$:  MOV      RHCS1(RO),SBDDAT ;GET THE CONTENTS OF RHCS1
(3) 053720 032737 020000 001126      BIT      #MCPE,SBDDAT   ;IS 'MCPE' SET ?
(3) 053726 001404          BEQ      .+12           ;BR IF NOT
(3) 053730 104011          ERROR    11            ;REPORT THE ERROR
(3) 053732 012760 040000 000000      MOV      #TRE,RHCS1(RO) ;CLEAR 'MCPE'
(3) 053740 005037 001236          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 053744 016037 000042 001126      MOV      RHER3(RO),SBDDAT ;GET CONTENTS OF RHER3
(3) 053752 012737 000042 001122      MOV      #RHER3,SBDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 053760 060037 001122          ADD      RO,SBDDADR    ;ADD RH11 BASE ADDRESS
(3) 053764 005037 001124          CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
(3) 053770 023737 001124 001126      CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(3) 053776 001403          BEQ      66$           ;BR IF OK
(3) 054000 104006          ERROR    6            ;TYPE MESSAGE 6
(3) 054002 005137 001236          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 054006 016037 000000 001126 66$:  MOV      RHCS1(RO),SBDDAT ;GET THE CONTENTS OF RHCS1
(3) 054014 032737 020000 001126      BIT      #MCPE,SBDDAT   ;IS 'MCPE' SET ?
(3) 054022 001404          BEQ      .+12           ;BR IF NOT
(3) 054024 104011          ERROR    11            ;REPORT THE ERROR
(3) 054026 012760 040000 000000      MOV      #TRE,RHCS1(RO) ;CLEAR 'MCPE'
(1) 054034 000207          RTS      PC            ;RETURN

```

6419  
6420

```

*****
*TEST 44      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 RHDS1.
*
*  B.  WRITE 1'S INTO RHER1, RHER2, & RHER3 THROUGH PORT 'B'.
*
*  C.  READ RHER1, RHER2, & RHER3 THROUGH PORT 'A'.  VERIFY THAT PORT
*      'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  D.  CLEAR RHER1, RHER2, & RHER3 THROUGH PORT 'B'.
*
*  E.  WRITE 1'S INTO RHER1, RHER2, & RHER3 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.
*
*****

```

```

(3) 054036 000004          TST44: SCOPE          ;INITIALIZE THE SCOPE HANDLER
(3) 054036 000004          TST      KYBCTL        ;PERFORMING ONLY SINGLE TESTS ?
(3) 054040 005737 001266      BEQ      2$            ;BR IF NOT
(3) 054044 001406          BPL      1$            ;BR IF JUST ENTERED TEST
(3) 054046 100002          JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
(3) 054050 000137 002410      MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
(3) 054054 012737 177777 001266 1$:

```

```

(3) 054062 112737 000044 001102 2S:   MOVB   #44,$STSTNM   ;TEST NUMBER
(3) 054070 012737 054112 001106   MOV   #TEST44,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 054076 012737 054112 001110   MOV   #TEST44,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 054104 012737 007640 001170   MOV   #4000,$TIMES   ;;DO 4000. ITERATIONS

```

6421  
5422

```

(3) ;:*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST

```

(2) 054112 TEST44:

(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(2) 054112 113760 001216 000010   MOVB   PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 054120 005060 000012   CLR    RHDS1(RO)       ;SEIZE THE DRIVE
(2) 054124 012760 000011 000000   MOV    #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 054132 012760 000013 000000   MOV    #13,RHCS1(RO)  ;RELEASE THE DRIVE
(2) 054140 113760 001220 000010   MOVB   PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 054146 005060 000012   CLR    RHDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 054152 012760 000011 000000   MOV    #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 054160 012760 000013 000000   MOV    #13,RHCS1(RO)  ;RELEASE THE DRIVE

```

(2) ;SEIZE THE DRIVE THROUGH PORT B

```

(2) 054166 113760 001220 000010   MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 054174 013737 001220 001230   MOV    PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
(2) 054202 005060 000012   CLR    RHDS1(RO)       ;WRITE RHDS1
(2) 054206 013737 001216 001232   MOV    PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
(2) 054214 012760 177777 000014   MOV    #-1,RHER1(RO)  ;LOAD 1'S INTO RHER1 THROUGH PORT B
(2) 054222 012760 177777 000040   MOV    #-1,RHER2(RO)  ;LOAD 1'S INTO RHER2 THROUGH PORT B
(2) 054230 012760 177777 000042   MOV    #-1,RHER3(RO)  ;LOAD 1'S INTO RHER3 THROUGH PORT B
(2) 054236 113760 001216 000010   MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 054244 013737 001216 001226   MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 054252 004737 055136   JSR    PC,TST44B      ;CHECK THE REGISTERS THROUGH PORT A
(2) 054256 113760 001220 000010   MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 054264 013737 001220 001226   MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 054272 005060 000042   CLR    RHER3(RO)      ;CLEAR RHER3 ON PORT B
(2) 054276 005060 000040   CLR    RHER2(RO)      ;CLEAR RHER2 ON PORT B
(2) 054302 005060 000014   CLR    RHER1(RO)      ;CLEAR RHER1 ON PORT B
(1) 054306 013760 001224 000016   MOV    ASR1,RHAS(RO)  ;CLEAR THE ATTENTION BIT FOR PORT B
(2) 054314 113760 001216 000010   MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
(2) 054322 013737 001216 001226   MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 054330 012760 177777 000014   MOV    #-1,RHER1(RO)  ;LOAD 1'S INTO RHER1 THROUGH PORT A
(2) 054336 012760 177777 000040   MOV    #-1,RHER2(RO)  ;LOAD 1'S INTO RHER2 THROUGH PORT A
(2) 054344 012760 177777 000042   MOV    #-1,RHER3(RO)  ;LOAD 1'S INTO RHER3 THROUGH PORT A
(2) 054352 113760 001220 000010   MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(2) 054360 013737 001220 001226   MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 054366 004737 055136   JSR    PC,TST44B      ;CHECK THE REGISTERS THROUGH PORT B

```

(2) ;RELEASE THE DRIVE FROM PORT B

```

(3) 054372 113760 001220 000010   MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
(3) 054400 013737 001220 001226   MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 054406 012760 000013 000000   MOV    #13,RHCS1(RO)  ;ISSUE RELEASE THROUGH PORT B

```

```

(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 054414 005037 001242 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 054420 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 054426 012737 000012 001122 MOV #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
(3) 054434 060037 001122 ADD R0,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 054440 113760 001216 000010 MOVB PORTA,RHCS2(R0) ;SELECT PORT A
(4) 054446 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 054454 016037 000012 001156 MOV RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
(4) 054462 113760 001220 000010 MOVB PORTB,RHCS2(R0) ;SELECT PORT B
(4) 054470 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 054476 016037 000012 001126 MOV RHDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT B
(3) 054504 001404 BEQ 64$ ;BR IF STATUS FROM PORT B ZERO
(3) 054506 005737 001156 TST $TMP0 ;IS STATUS FROM PORT A ZERO ?
(3) 054512 001401 BEQ 64$ ;BR IF ZERO
(3) 054514 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 054516 013737 001156 001126 64$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
(3) 054524 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 054532 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 054540 001401 BEQ +4 ;BR IF OK
(3) 054542 104027 ERROR 27 ;REPORT REGISTER ERROR

(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(2)
(3) 054544 113760 001216 000010 MOVB PORTA,RHCS2(R0) ;SELECT PORT A
(3) 054552 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 054560 012760 000013 000000 MOV #13,RHCS1(R0) ;ISSUE RELEASE THROUGH PORT A

(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 054566 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 054572 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 054600 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
(3) 054604 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 054612 113760 001216 000010 MOVB PORTA,RHCS2(R0) ;SELECT PORT A.
(3) 054620 016037 000012 001162 MOV RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 054626 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 054634 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 054642 113760 001220 000010 MOVB PORTB,RHCS2(R0) ;SELECT PORT B.
(3) 054650 016037 000012 001164 MOV RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 054656 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 054664 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 054672 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 054700 001006 BNE 65$ ;BR IF NOT
(3) 054702 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 054706 001045 BNE 67$ ;BR IF NOT
(3) 054710 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 054712 000137 055076 JMP 69$ ;BYPASS THE REST OF THE CHECKS
(3) 054716 013737 001162 001126 65$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 054724 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 054732 113760 001220 000010 MOVB PORTB,RHCS2(R0) ;SELECT PORT B.
(3) 054740 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 054744 001414 BEQ 66$ ;BR IF ZERO

```

```

(3) 054746 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 054754 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 054762 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 054770 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 054774 001012 BNE 67$ ;BR IF NOT
(3) 054776 012737 177777 001242 66$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 055004 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 055012 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 055020 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 055022 013737 001162 001126 67$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 055030 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 055036 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 055044 001401 BEQ 68$ ;BR IF OK FROM PORT A.
(3) 055046 104007 ERROR 7 ;REPORT ERROR
(3) 055050 013737 001164 001126 68$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 055056 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 055064 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 055072 001401 BEQ 69$ ;BR IF OK
(3) 055074 104007 ERROR 7 ;REPORT ERROR
(3) 055076 000240 69$: NOP

(2) ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
(2)
(2)
(2) 055100 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
(4) 055104 001547 BEQ TST45 ;:BR IF NOT
(2) 055106 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
(4) 055114 001543 BEQ TST45 ;:BR IF NOT
(2) 055116 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
(2) 055122 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
(2) 055126 000177 123756 JMP @SLPERR ;GO TO THE LOOP ADDRESS
6423 055132 000137 055424 JMP TST45 ;GO TO THE NEXT TEST

(1) ;CHECK THE REGISTERS ON THE SELECTED PORT
(1)
(1)
(1) 055136 TST44B:
(3) 055136 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 055142 016037 000014 001126 MOV RHER1(RO),$BDDAT ;GET CONTENTS OF RHER1
(3) 055150 012737 000014 001122 MOV #RHER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 055156 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(3) 055162 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(3) 055166 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 055174 001403 BEQ 64$ ;BR IF OK
(3) 055176 104006 ERROR 6 ;TYPE MESSAGE 6
(3) 055200 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 055204 016037 000000 001126 64$: MOV RHCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
(3) 055212 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 055220 001404 BEQ +12 ;BR IF NOT
(3) 055222 104011 ERROR 11 ;REPORT THE ERROR
(3) 055224 012760 040000 000000 MOV #TRE,RHCS1(RO) ;CLEAR 'MCPE'
(3) 055232 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 055236 016037 000040 001126 MOV RHER2(RO),$BDDAT ;GET CONTENTS OF RHER2
(3) 055244 012737 000040 001122 MOV #RHER2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 055252 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(3) 055256 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE

```

```

(3) 055262 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
(3) 055270 001403                    BEQ      65$                ;BR IF OK
(3) 055272 104006                    ERROR    6                  ;TYPE MESSAGE 6
(3) 055274 005137 001236            COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 055300 016037 000000 001126 65$: MOV      RHCS1(RO), $BDDAT    ;GET THE CONTENTS OF RHCS1
(3) 055306 032737 020000 001126      BIT      #MCPE,$BDDAT      ;IS 'MCPE' SET ?
(3) 055314 001404                    BEQ      +12                ;BR IF NOT
(3) 055316 104011                    ERROR    11                 ;REPORT THE ERROR
(3) 055320 012760 040000 000000      MOV      #TRE,RHCS1(RO)    ;CLEAR 'MCPE'
(3) 055326 005037 001236            CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 055332 016037 000042 001126      MOV      #RHER3,$BDDAT    ;GET CONTENTS OF RHER3
(3) 055340 012737 000042 001122      MOV      #RHER3,$BADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 055346 060037 001122            ADD      RO,$BADR         ;ADD RH11 BASE ADDRESS
(3) 055352 005037 001124            CLR      $GDDAT           ;WHAT REGISTER SHOULD BE
(3) 055356 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
(3) 055364 001403                    BEQ      66$                ;BR IF OK
(3) 055366 104006                    ERROR    6                  ;TYPE MESSAGE 6
(3) 055370 005137 001236            COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 055374 016037 000000 001126 66$: MOV      RHCS1(RO), $BDDAT    ;GET THE CONTENTS OF RHCS1
(3) 055402 032737 020000 001126      BIT      #MCPE,$BDDAT    ;IS 'MCPE' SET ?
(3) 055410 001404                    BEQ      +12                ;BR IF NOT
(3) 055412 104011                    ERROR    11                 ;REPORT THE ERROR
(3) 055414 012760 040000 000000      MOV      #TRE,RHCS1(RO)   ;CLEAR 'MCPE'
(1) 055422 000207                    RTS      PC                 ;RETURN

```

6424  
6437  
6438

```

*****
;TEST 45      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 RHDS1.
;
; C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
;    FOR THE DRIVE IS SET.
;
*****

```

```

(3) 055424 000004                    TST45: SCOPE                ;INITIALIZE THE SCOPE HANDLER
(3) 055424 005737 001266            TST      KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
(3) 055426 001406                    BEQ      2$                ;BR IF NOT
(3) 055432 100002                    BPL      1$                ;BR IF JUST ENTERED TEST
(3) 055436 000137 002410            JMP      EXEC              ;RETURN & GET NEXT TEST NUMBER
(3) 055442 012737 177777 001266 1$: MOV      #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
(3) 055450 112737 000045 001102 2$: MOVB    #45,$STNM        ;TEST NUMBER
(3) 055456 012737 055500 001106      MOV      #TEST45,$LPADR   ;LOAD LOOP ON TEST ADDRESS
(3) 055464 012737 055500 001110      MOV      #TEST45,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
(1) 055472 012737 007640 001170      MOV      #4000,$TIMES    ;DO 4000. ITERATIONS

```

6439  
6480  
(3)  
(2)

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
*****

```

```

(2)
(2) 055500          TEST45:
(2)
(2)                ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 055500 113760 001216 000010  MOVB  PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 055506 005060 000012          CLR   RHDS1(RO)       ;SEIZE THE DRIVE
(2) 055512 012760 000011 000000  MOV   #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 055520 012760 000013 000000  MOV   #13,RHCS1(RO)  ;RELEASE THE DRIVE
(2) 055526 113760 001220 000010  MOVB  PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 055534 005060 000012          CLR   RHDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 055540 012760 000011 000000  MOV   #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 055546 012760 000013 000000  MOV   #13,RHCS1(RO)  ;RELEASE THE DRIVE
(1) 055554 113760 001216 000010  MOVB  PORTA,RHCS2(RO) ;SELECT PORT A
(1) 055562 012760 177777 000014  MOV   #-1,RHER1(RO)  ;SET ERRORS TO FORCE ATTN BIT ON PORT A
(1) 055570 005060 000014          CLR   RHER1(RO)      ;CLEAR THE ERRORS
(1) 055574 113760 001220 000010  MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(1) 055602 005760 000012          1$:  TST   RHDS1(RO)      ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
(1) 055606 001775          BEQ   1$             ;BR IF STILL SEIZED BY PORT A
(1) 055610 012737 000016 001122  MOV   #RHAS,$BDADR   ;FORM ADDRESS OF ATTN REG IF ERROR
(1) 055616 060037 001122          ADD   RO,$BDADR      ;ADD THE ADDRESS BASE
(1) 055622 013737 001224 001124  MOV   ASR1,$GDDAT    ;GOOD DATA FOR ERROR MESSAGE
(1) 055630 013737 001224 001160  MOV   ASR1,$TMP1     ;MAKE DATA COMPARE MASK
(1) 055636 005137 001160          COM   $TMP1          ;COMPLEMENT IT
(1) 055642 012737 000102 001110  MOV   #'B,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
(2) 055650 113760 001220 000010  MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(2) 055656 013737 001220 001226  MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 055664 013737 001220 001230  MOV   PORTB,SEIZPT   ;'SEIZED' PORT ADDRESS
(1) 055672 005060 000012          CLR   RHDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT B
(1) 055676 016037 000016 001126  2$:  MOV   RHAS(RO),$BDAT ;GET THE CONTENTS OF THE ATTENTION REG
(1) 055704 013737 001126 001156  MOV   $BDAT,$TMP0    ;PUT CONTENTS INTO WORKING LOCATION
(1) 055712 043737 001160 001156  BIC   $TMP1,$TMP0    ;CLEAR OTHER BITS
(1) 055720 023737 001124 001156  CMP   $GDDAT,$TMP0   ;SEE IF ATTN BIT FOR DRIVE SET
(1) 055726 001411          BEQ   3$             ;BR IF SET
(1) 055730 104053          ERROR 53           ;REPORT THE ERROR
(1) 055732 005037 001170          CLR   $TIMES        ;CLEAR ITERATION COUNT
(1) 055736 032737 001000 177570  BIT   #SW09,$SWR     ;LOOP ON THE ERROR ?
(1) 055744 001402          BEQ   .+6           ;BR IF SW09 NOT SET
(1) 055746 000177 123136          JMP   2$LPERR       ;GO TO THE LOOP ADDRESS
(1) 055752 005237 001104          3$:  INC   $ICNT          ;INCREMENT THE ITERATION COUNT
(1) 055756 023737 001104 001170  CMP   $ICNT,$TIMES  ;DO THE SUBTEST SOME MORE ?
(1) 055764 002002          BGE   .+6           ;BR IF NOT
(1) 055766 000137 055676          JMP   2$            ;DO THE SUBTEST AGAIN
(2)
(2)                ;RELEASE THE DRIVE FROM PORT B
(2)
(2) 055772 113760 001220 000010  MOVB  PORTB,RHCS2(RO) ;SELECT PORT B
(2) 056000 013737 001220 001226  MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 056006 012760 000013 000000  MOV   #13,RHCS1(RO)  ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 056014 005037 001242          CLR   RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 056020 012737 000012 001122  MOV   #RHDS1,$BDADR  ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT

```



```

(3) 056026 060037 001122          ADD      RD,$BDADR          ;ADD THE I/O BASE ADDRESS
(3) 056032 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!V,SGDDAT ;COMPARISON CONSTANT
(3) 056040 113760 001216 000010  MOV      PORTA,RHCS2(RD) ;SELECT PORT A.
(3) 056046 016037 000012 001162  MOV      RHDS1(RD),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 056054 013737 001162 001156  MOV      STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 056062 042737 100100 001156  BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 056070 113760 001220 000010  MOV      PORTB,RHCS2(RD) ;SELECT PORT B.
(3) 056076 016037 000012 001164  MOV      RHDS1(RD),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 056104 013737 001164 001160  MOV      STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 056112 042737 100100 001160  BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 056120 023737 001156 001160  CMP      STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 056126 001006          BNE      64$ ;BR IF NOT
(3) 056130 005737 001156          TST      STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 056134 001045          BNE      66$ ;BR IF NOT
(3) 056136 104046          ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 056140 000137 056340          JMP      68$ ;BYPASS THE REST OF THE CHECKS
(3) 056144 013737 001162 001126 64$: MOV      STMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 056152 013737 001220 001226  MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 056160 113760 001220 000010  MOV      PORTB,RHCS2(RD) ;SELECT PORT B.
(3) 056166 005737 001156          TST      STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 056172 001414          BEQ      65$ ;BR IF ZERO
(3) 056174 013737 001216 001226  MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 056202 013737 001164 001126  MOV      STMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 056210 113760 001216 000010  MOV      PORTA,RHCS2(RD) ;SELECT PORT A.
(3) 056216 005737 001160          TST      STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 056222 001012          BNE      66$ ;BR IF NOT
(3) 056224 012737 177777 001242 65$: MOV      #-1,RELEERR ;SET 'RELEASE ERROR' INDICATOR
(3) 056232 012760 000011 000000  MOV      #11,RHCS1(RD) ;CLEAR THE DRIVE
(3) 056240 012760 000013 000000  MOV      #13,RHCS1(RD) ;RELEASE THE DRIVE
(3) 056246 104026          ERROR    26 ;TYPE ERROR MESSAGE 26
(3) 056250 013737 001162 001126 66$: MOV      STMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 056256 013737 001216 001226  MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 056264 042737 100000 001162  BIC      #ATA,STMP2 ;DON'T CHECK THE ATTN BIT
(3) 056272 023737 001124 001162  CMP      SGDDAT,STMP2 ;ALL BITS OK ?
(3) 056300 001401          BEQ      67$ ;BR IF OK FROM PORT A.
(3) 056302 104007          ERROR    7 ;REPORT ERROR
(3) 056304 013737 001164 001126 67$: MOV      STMP3,$BDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 056312 013737 001220 001226  MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 056320 042737 100000 001164  BIC      #ATA,STMP3 ;DON'T CHECK THE ATTN BIT
(3) 056326 023737 001124 001164  CMP      SGDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 056334 001401          BEQ      68$ ;BR IF OK
(3) 056336 104007          ERROR    7 ;REPORT ERROR
(3) 056340 000240          NOP      7

```

6493  
6494  
(3)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)  
(4)

```

;*****
;TEST 46 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 RHDS1.
;
; C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT

```

```

(4) ;* FOR THE DRIVE IS SET.
(4) ;*
(3) ;*****
(3) 056342 TST46: SCOPE ;INITIALIZE THE SCOPE HANDLER
(3) 056342 000004 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 056344 005737 001266 BEQ 2$ ;BR IF NOT
(3) 056350 001406 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 056352 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 056354 000137 002410 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 056360 012737 177777 001266 2$: MOV#B #46,$TSTNM ;TEST NUMBER
(3) 056366 112737 000046 001102 MOV #TEST46,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 056374 012737 056416 001106 MOV #TEST46,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 056402 012737 056416 001110 MOV #4000.,$TIMES ;DO 4000. ITERATIONS
(1) 056410 012737 007640 001170
6495
6496
(3) ;*****
(2) ;END OF 'SCOPE' SETUP - START OF MAIN TEST
(2) TEST46:
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 056416
(2) MOV#B PORTA,RHCS2(RO) ;SELECT PORT #A
(2) 056424 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
(2) 056430 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 056436 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(2) 056444 113760 001220 000010 MOV#B PORTB,RHCS2(RO) ;SELECT PORT #B
(2) 056452 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 056456 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
(2) 056464 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(1) 056472 113760 001220 000010 MOV#B PORTB,RHCS2(RO) ;SELECT PORT B
(1) 056500 012760 177777 000014 MOV #-1,RHER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT B
(1) 056506 005060 000014 CLR RHER1(RO) ;CLEAR THE ERRORS
(1) 056512 113760 001216 000010 MOV#B PORTA,RHCS2(RO) ;SELECT PORT A
(1) 056520 005760 000012 1$: TST RHDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
(1) 056524 001775 BEQ 1$ ;BR IF STILL SEIZED BY PORT B
(1) 056526 012737 000016 001122 MOV #RHAS,$BDADR ;FORM ADDRESS OF ATTN REG IF ERROR
(1) 056534 060037 001122 ADD RO,$BDADR ;ADD THE ADDRESS BASE
(1) 056540 013737 001224 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
(1) 056546 013737 001224 001160 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
(1) 056554 005137 001160 COM $TMP1 ;COMPLEMENT IT
(1) 056560 012737 000102 001110 MOV #'B,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(2) 056566 113760 001216 000010 MOV#B PORTA,RHCS2(RO) ;SELECT PORT A
(2) 056574 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 056602 013737 001216 001230 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
(1) 056610 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT A
(1) 056614 016037 000016 001126 2$: MOV RHAS(RO),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
(1) 056622 013737 001126 001156 MOV $BDDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
(1) 056630 043737 001160 001156 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
(1) 056636 023737 001124 001156 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
(1) 056644 001411 BEQ 3$ ;BR IF SET
(1) 056646 104053 ERROR 53 ;REPORT THE ERROR
(1) 056650 005037 001170 CLR $TIMES ;CLEAR ITERATION COUNT

```

```

(1) 056654 032737 001000 177570 BIT #SW09,SWR ;LOOP ON THE ERROR ?
(1) 056662 001402 BEQ .+6 ;BR IF SW09 NOT SET
(1) 056664 000177 122220 JMP @SLPERR ;GO TO THE LOOP ADDRESS
(1) 056670 005237 001104 3$: INC $ICNT ;INCREMENT THE ITERATION COUNT
(1) 056674 023737 001104 001170 CMP $ICNT,$TIMES ;DO THE SUBTEST SOME MORE ?
(1) 056702 002002 BGE .+6 ;BR IF NOT
(1) 056704 000137 056614 JMP 2$ ;DO THE SUBTEST AGAIN
(2)
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 056710 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
(3) 056716 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 056724 012760 000013 000000 MOV #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 056732 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 056736 012737 000012 001122 MOV #RHDS1,$BDDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
(3) 056744 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(3) 056750 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 056756 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 056764 016037 000012 001162 MOV RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 056772 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 057000 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 057006 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 057014 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 057022 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 057030 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 057036 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 057044 001006 BNE 64$ ;BR IF NOT
(3) 057046 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 057052 001045 BNE 66$ ;BR IF NOT
(3) 057054 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 057056 000137 057256 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(3) 057062 013737 001162 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 057070 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 057076 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
(3) 057104 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 057110 001414 BEQ 65$ ;BR IF ZERO
(3) 057112 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 057120 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 057126 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
(3) 057134 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 057140 001012 BNE 66$ ;BR IF NOT
(3) 057142 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 057150 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
(3) 057156 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
(3) 057164 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 057166 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
(3) 057174 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 057202 042737 100000 001162 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(3) 057210 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 057216 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(3) 057220 104007 ERROR 7 ;REPORT ERROR

```

```

(3) 057222 013737 001164 001126 67S: MOV $TMP3,$BCDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
(3) 057230 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 057236 042737 100000 001164 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 057244 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 057252 001401 BEQ 68S ;BR IF OK
(3) 057254 104007 ERROR 7 ;REPORT ERROR
(3) 057256 000240 68S: NOP
6497 057260 000137 057510 JMP SEOP ;GO TO THE END OF PASS ROUTINE

```

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

.SBTTL \*\*\* SUBROUTINES \*\*\*

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

;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS  
;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED

```

6509 057264 012737 057334 000004 CKCLK: MOV #CKCLK1,2#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
6510 057272 005037 000006 CLR 2#ERRVEC+2 ;NEW PSW
6511 057276 005777 121702 TST 2$SLKCSR ;CHECK FOR KW11-P
6512 057302 013701 001210 MOV $SLPVEC,R1 ;KW11-P VECTOR ADDRESS
6513 057306 012721 057416 MOV #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
6514 057312 012711 000300 MOV #300,(R1) ;PSW - PRI 6
6515 057316 012777 177777 121662 MOV #-1,2$SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
6516 057324 012777 000135 121652 MOV #135,2$SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
6517 057332 000425 BR CKCLK3
6518 057334 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
6519 057340 012737 057376 000004 MOV #CKCLK2,2#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
6520 057346 005777 121640 TST 2$SLKS ;LOOK FOR KW11-L
6521 057352 013701 001214 MOV $LLVEC,R1 ;KW11-L VECTOR ADDRESS
6522 057356 012721 057416 MOV #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
6523 057362 012711 000300 MOV #300,(R1) ;PSW - PRI 6
6524 057366 012777 000100 121616 MOV #100,2$SLKS ;SET KW11-L INTERRUPT
6525 057374 000404 BR CKCLK3
6526 057376 062706 000004 CKCLK2: ADD #4,SP ;RESTORE THE STACK POINTER
6527 057402 062716 000002 ADD #2,(SP) ;INCREMENT RETURN, NO CLOCK
6528 057406 012737 000006 000004 CKCLK3: MOV #6,2#ERRVEC ;RESTORE THE ERROR VECTOR
6529 057414 000207 RTS PC

```

;ROUTINE TO COUNT CLOCK TICKS

```

6533 057416 062737 000021 001244 CLOCK: ADD #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
6534 057424 005737 001246 TST WATCH ;IS WATCH ALREADY ZERO ?
6535 057430 001406 BEQ 1$ ;BR IF IT IS
6536 057432 162737 000021 001246 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
6537 057440 100002 BPL 1$ ;BR IF NOT MINUS
6538 057442 005037 001246 CLR WATCH ;CLEAR WATCH DOG COUNTER
6539 057446 000002 1$: RTI ;RETURN

```

;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES

```

6543 057450 162706 000004 TOLER: SUB #4,SP ;SETUP STACK

```

```

6544 057454 016616 000004      MOV      4(SP), (SP)      ;SAVE STACK
6545 057460 013546              MOV      2(R5)+, -(SP)   ;GET TIME VALUE
6546 057462 011666 000004      MOV      (SP), 4(SP)    ;MOVE TIME VALUE
6547 057466 011666 000006      MOV      (SP), 6(SP)    ;MOVE VALUE AGAIN
6548 057472 006216              ASR      (SP)            ;DIVIDE BY 2
6549 057474 006216              ASR      (SP)            ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
6550 057476 061666 000004      ADD      (SP), 4(SP)    ;CALCULATE UPPER LIMIT FOR TIMEOUT
6551 057502 162666 000004      SUB      (SP)+, 4(SP)   ;CALCULATE LOWER LIMIT FOR TIMEOUT
6552 057506 000205      RTS      R5              ;RETURN WITH TOLERANCES ON THE STACK
6553
6554 ;*****
6555 .SBTTL 'SYSMAC' UTILITY ROUTINES
6556 ;*****
6557 ;*****
6558 ;*****
6559 ;*****
6560
(1) .SBTTL END OF PASS ROUTINE
(1)
(1)
(1)
(1) ;*INCREMENT THE PASS NUMBER ($PASS)
(1) ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
(1) ;*TYPE "END PASS #XXXX" (WHERE XXXX IS A DECIMAL NUMBER)
(1) ;*IF THERES A MONITOR GO TO IT
(1) ;*IF THERE ISN'T JUMP TO TSTIAA
(1)
(1) SEOP:
(1) 057510              SCOPE
(3) 057510 000004              TST      KYBCTL          ;ENTERED TEST VIA KEYBOARD COMMAND ?
(3) 057512 005737 001266      BEQ      .+6             ;BR IF NOT
(3) 057516 001402              JMP      EXEC            ;RETURN TO KEYBOARD CONTROL
(1) 057520 000137 002410      CLR      $STNM          ;ZERO THE TEST NUMBER
(1) 057524 005037 001102      CLR      $TIMES         ;ZERO THE NUMBER OF ITERATIONS
(1) 057530 005037 001170      INC      $PASS          ;INCREMENT THE PASS NUMBER
(1) 057534 005237 001100      BIC      #100000, $PASS ;DON'T ALLOW A NEG. NUMBER
(1) 057540 042737 100000 001100      DEC      (PC)+          ;LOOP?
(1) 057546 005327              SEOPCT: .WORD           1
(1) 057550 000001              BGT      $DOAGN         ;YES
(1) 057552 003031              MOV      (PC)+, 2(PC)+ ;RESTORE COUNTER
(1) 057554 012737              SENDCT: .WORD           1
(1) 057556 000001              SEOPCT
(1) 057560 057550              TYPE      SENDMG        ;TYPE "END PASS #"
(1) 057562 104400 057642      MOV      $PASS, -(SP)   ;SAVE $PASS FOR TYPEOUT
(2) 057566 013746 001100      TYPDS    ;GO TYPE--DECIMAL ASCII WITH SIGN
(2) 057572 104410              TYPE      $NULL         ;TYPE A NULL CHARACTER
(1) 057574 104400 057657      MOV      2#42, R0       ;GET MONITOR ADDRESS
(1) 057600 013700 000042      BEQ      $DOAGN         ;BRANCH IF NO MONITOR
(1) 057604 001414              CMP      #SENDAD, R0    ;IS MONITOR ACT11?
(1) 057606 022700 057626      BNE      SENDAD         ;NO--BRANCH
(1) 057612 001005              CMP      #-1, 2(R0)     ;YES--IS THIS THE LAST PASS?
(1) 057614 022760 177777 000002      BNE      $DOAGN         ;NO--MAKE ANOTHER PASS
(1) 057622 001005              RESET
(1) 057624 000005              SENDAD: JSR      PC, (R0) ;CLEAR THE WORLD
(1) 057626 004710              NOP                    ;GO TO MONITOR
(1) 057630 000240              ;SAVE ROOM

```

```

(1) 057632 000240      NOP                ;;FOR
(1) 057634 000240      NOP                ;;ACT11
(1) 057636 000137 002704  SDOAGN: JMP      @#TST1AA      ;;RETURN
(1) 057642 005015 047105 020104 SENDMG: .ASCIZ  <15><12>/END PASS #/
(1) 057650 040520 051523 021440
(1) 057656      000
(1) 057657      377      377      000 $ENULL: .BYTE  -1,-1,0      ;;NULL CHARACTER STRING
6570 ;*****
(1) .SBTTL  SCOPE HANDLER ROUTINE
(1)
(1) ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1) ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1) ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1) ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;*SW14=1      LOOP ON TEST
(1) ;*SW11=1      INHIBIT ITERATIONS
(1) ;*CALL
(1) ;*      SCOPE      ;;SCOPE=IOT
(1)
(1) 057662      $SCOPE:
(1) 057662 006137 177570      ROL      @#SWR      ;;LOOP ON PRESENT TEST?
(1) 057666 100455      BMI      $OVER     ;;YES IF SW14=1
(1) ;*****START OF CODE FOR THE XOR TESTER*****
(1) 057670 000416      $XTSTR: BR      6$      ;; IF RUNNING ON THE "XOR" TESTER CHANGE
(1) ;*THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 057672 013746 000004      MOV      @#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 057676 012737 057716 000004      MOV      #5$,@#ERRVEC  ;;SET FOR TIMEOUT
(1) 057704 005737 177060      TST     @#177060      ;;TIME OUT ON XOR?
(1) 057710 012637 000004      MOV      (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
(1) 057714 000436      BR      $SVLAD      ;;GO TO THE NEXT TEST
(1) 057716 022626      5$: CMP     (SP)+,(SP)+  ;;CLEAR THE STACK AFTER A TIME OUT
(1) 057720 012637 000004      MOV      (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
(1) 057724 000436      BR      $OVER      ;;LOOP ON THE PRESENT TEST
(1) 057726      6$;*****END OF CODE FOR THE XOR TESTER*****
(1) 057726 105737 001103      2$: TSTB  $ERFLG      ;;HAS AN ERROR OCCURRED?
(1) 057732 001404      BEQ     3$          ;;BR IF NO
(1) 057734 105037 001103      4$: CLRB  $ERFLG      ;;ZERO THE ERROR FLAG
(1) 057740 005037 001170      CLR     $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 057744 032737 004000 177570 3$: BIT   @BIT11,@#SWR  ;;INHIBIT ITERATIONS?
(1) 057752 001011      BNE     1$          ;;BR IF YES
(1) 057754 005737 001100      TST     $PASS      ;;IF FIRST PASS OF PROGRAM
(1) 057760 001406      BEQ     1$          ;;INHIBIT ITERATIONS
(1) 057762 005237 001104      INC     $ICNT      ;;INCREMENT ITERATION COUNT
(1) 057766 023737 001170 001104      CMP     $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
(1) 057774 002012      BGE     $OVER      ;;BR IF MORE ITERATION REQUIRED
(1) 057776 012737 000001 001104 1$: MOV   @1,$ICNT      ;;REINITIALIZE THE ITERATION COUNTER
(1) 060004 013737 060036 001170      MOV   $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
(1) 060012 105237 001102      $SVLAD: INCB  $STNM   ;;COUNT TEST NUMBERS
(1) 060016 011637 001106      MOV   (SP),$LPADR  ;;SAVE SCOPE LOOP ADDRESS
(1) 060022 013737 001102 177570 $OVER: MOV   $STNM,@#DISPLAY ;;DISPLAY TEST NUMBER
(1) 060030 013716 001106      MOV   $LPADR,(SP)  ;;FUDGE RETURN ADDRESS
(1) 060034 000002      RTI
(1) 060036 000004      $MXCNT: 4      ;;MAX. NUMBER OF ITERATIONS

```

```

6571 ;*****
(1) .SBTTL ERROR HANDLER ROUTINE
(1) ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1) ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1) ;*AND GO TO SERRTYP ON ERROR
(1) ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;*SW15=1 HALT ON ERROR
(1) ;*SW13=1 INHIBIT ERROR TYPEOUTS
(1) ;*SW10=1 BELL ON ERROR
(1) ;*CALL
(1) ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1) SERROR:
(1) 060040 113737 001102 001234 MOVB $TSTNM,TSTNUM
(2) 060040 105237 001103 7$: INCB $ERFLG ;;SET THE ERROR FLAG
(1) 060052 001775 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
(1) 060054 013737 001102 177570 MOV $TSTNM,$DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1) 060062 032737 002000 177570 BIT $BIT10,$SWR ;;BELL ON ERROR?
(1) 060070 001402 BEQ 1$ ;;NO - SKIP
(1) 060072 104400 001174 TYPE $BELL ;;RING BELL
(1) 060076 005237 001112 1$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
(1) 060102 011637 001116 MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
(1) 060106 162737 000002 001116 SUB #2,$ERRPC
(1) 060114 117737 120776 001114 MOVB $ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
(1) 060122 032737 020000 177570 BIT $BIT13,$SWR ;;SKIP TYPEOUT IF SET
(1) 060130 001004 BNE 2$ ;;SKIP TYPEOUTS
(1) 060132 004737 060154 JSR PC,$SERRTYP ;;GO TO USER ERROR ROUTINE
(1) 060136 104400 001201 TYPE $SCLF
(1) 060142 005737 177570 2$: TST $SWR ;;HALT ON ERROR
(1) 060146 100001 BPL 3$ ;;SKIP IF CONTINUE
(1) 060150 000000 HALT ;;HALT ON ERROR!
(1) 060152 3$:
(1) 060152 000002 RTI ;;RETURN

```

```

6572 ;*****
(1) .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
(1) ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1) ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1) ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1) SERRTYP:
(1) 060154 104400 001201 TYPE $SCLF ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 060160 010046 MOV RO,-(SP) ;;SAVE RO
(1) 060162 005000 CLR RO ;;PICKUP THE ITEM INDEX
(1) 060164 153700 001114 BISB $ITEMB,RO
(1) 060170 001004 BNE 1$ ;;IF ITEM NUMBER IS ZERO, JUST
(1) MOV $ERRPC,-(SP) ;;TYPE THE PC OF THE ERROR
(2) 060172 013746 001116 ;;SAVE $ERRPC FOR TYPEOUT
(2) ;;ERROR ADDRESS
(1) 060176 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 060200 000445 BR 10$ ;;GET OUT

```

```

(1) 060202 005300          1$: DEC      RO          ;; ADJUST THE INDEX SO THAT IT WILL
(1) 060204 006300          ASL      RO          ;; WORK FOR THE ERROR TABLE
(1) 060206 006300          ASL      RO
(1) 060210 006300          ASL      RO
(1) 060212 062700 001274  ADD      #ERRTB,RO  ;; FORM TABLE POINTER
(1) 060216 012037 060226  MOV      (RO)+,2$  ;; PICKUP "ERROR MESSAGE" POINTER
(1) 060222 001404          BEQ      3$          ;; SKIP TYPEOUT IF NO POINTER
(1) 060224 104400          TYPE     ;; TYPE THE "ERROR MESSAGE"
(1) 060226 000000          .WORD   0          ;; "ERROR MESSAGE" POINTER GOES HERE
(1) 060230 104400 001201  TYPE     ,SCLF     ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 060234 012037 060244  MOV      (RO)+,4$  ;; PICKUP "DATA HEADER" POINTER
(1) 060240 001404          BEQ      5$          ;; SKIP TYPEOUT IF 0
(1) 060242 104400          TYPE     ;; TYPE THE "DATA HEADER"
(1) 060244 000000          .WORD   0          ;; "DATA HEADER" POINTER GOES HERE
(1) 060246 104400 001201  TYPE     ,SCLF     ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 060252 010146          MOV      R1,-(SP)  ;; SAVE R1
(1) 060254 012001          MOV      (RO)+,R1  ;; PICKUP "DATA TABLE" POINTER
(1) 060256 001415          BEQ      9$          ;; BR IF NO DATA TO BE TYPED
(1) 060260 012000          MOV      (RO)+,RO  ;; PICKUP "DATA FORMAT" POINTER
(1) 060262 105720          6$: TSTB    (RO)+     ;; "OCTAL" OR "DECIMAL"
(1) 060264 001003          BNE     7$          ;; BR IF DECIMAL
(2) 060266 013146          MOV      2(R1)+,-(SP) ;; SAVE 2(R1)+ FOR TYPEOUT
(2) 060270 104402          TYPOC   ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 060272 000402          BR      8$
(2) 060274          7$:
(2) 060274 013146          MOV      2(R1)+,-(SP) ;; SAVE 2(R1)+ FOR TYPEOUT
(2) 060276 104410          TYPDS   ;; GO TYPE--DECIMAL ASCII WITH SIGN
(1) 060300 005711          8$: TST     (R1)     ;; IS THERE ANOTHER NUMBER?
(1) 060302 001403          BEQ     9$          ;; BR IF NO
(1) 060304 104400 060324  TYPE     ,11$     ;; TYPE TWO(2) SPACES
(1) 060310 000764          BR      6$          ;; LOOP
(1) 060312 012601          9$: MOV     (SP)+,R1  ;; RESTORE R1
(1) 060314 012600          10$: MOV    (SP)+,RO ;; RESTORE RO
(1) 060316 104400 001201  TYPE     ,SCLF     ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 060322 000207          RTS     PC         ;; RETURN
(1) 060324 020040 000          11$: .ASCIZ  / /      ;; TWO(2) SPACES
(1) 060330 060330          .EVEN

```

6573 ;\*\*\*\*\*

(1) .SBTTL TYPE ROUTINE

```

(1) ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1) ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1) ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1) ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1) ;*
(1) ;*CALL:
(1) ;*1) USING A TRAP INSTRUCTION
(1) ;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1) ;*OR
(1) ;* TYPE
(1) ;* MESADR

```





```

(1)          ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1)          ;*      TYPOC      ;;CALL FOR TYPEOUT
(1)
(1) 060436 017646 000000          $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
(1) 060442 116637 000001 060661  MOVB     1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
(1) 060450 112637 060663          MOVB     (SP)+,$OMODE+1      ;;NUMBER OF DIGITS TO TYPE
(1) 060454 062716 000002          ADD      #2,(SP)          ;;ADJUST RETURN ADDRESS
(1) 060460 000406          BR       $TYPON
(1) 060462 112737 000001 060661  $TYPOC: MOVB     #1,$OFILL      ;;SET THE ZERO FILL SWITCH
(1) 060470 112737 000006 060663  MOVB     #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
(1) 060476 112737 000005 060660  $TYPON: MOVB     #5,$OCNT      ;;SET THE ITERATION COUNT
(1) 060504 010346          MOV      R3,-(SP)        ;;SAVE R3
(1) 060506 010446          MOV      R4,-(SP)        ;;SAVE R4
(1) 060510 010546          MOV      R5,-(SP)        ;;SAVE R5
(1) 060512 113704 060663          MOVB     $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
(1) 060516 005404          NEG      R4
(1) 060520 062704 000006          ADD      #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
(1) 060524 110437 060662          MOVB     R4,$OMODE        ;;SAVE IT FOR USE
(1) 060530 113704 060661          MOVB     $OFILL,R4        ;;GET THE ZERO FILL SWITCH
(1) 060534 016605 000012          MOV      12(SP),R5       ;;PICKUP THE INPUT NUMBER
(1) 060540 005003          CLR      R3              ;;CLEAR THE OUTPUT WORD
(1) 060542 006105          1$:     ROL      R5        ;;ROTATE MSB INTO "C"
(1) 060544 000404          BR       3$
(1) 060546 006105          2$:     ROL      R5        ;;FORM THIS DIGIT
(1) 060550 006105          ROL      R5
(1) 060552 006105          ROL      R5
(1) 060554 010503          MOV      R5,R3
(1) 060556 006103          3$:     ROL      R3        ;;GET LSB OF THIS DIGIT
(1) 060560 105337 060662          DECB     $OMODE          ;;TYPE THIS DIGIT?
(1) 060564 100016          BPL      7$              ;;BR IF NO
(1) 060566 042703 177770          BIC      #177770,R3      ;;GET RID OF JUNK
(1) 060572 001002          BNE      4$              ;;TEST FOR 0
(1) 060574 005704          TST      R4              ;;SUPPRESS THIS 0?
(1) 060576 001403          BEQ      5$              ;;BR IF YES
(1) 060600 005204          4$:     INC      R4        ;;DON'T SUPPRESS ANYMORE 0'S
(1) 060602 052703 000060          BIS      #'0,R3          ;;MAKE THIS DIGIT ASCII
(1) 060606 052703 000040          5$:     BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
(1) 060612 110337 060656          MOVB     R3,8$          ;;SAVE FOR TYPING
(1) 060616 104400 060656          TYPE     8$              ;;GO TYPE THIS DIGIT
(1) 060622 105337 060660          7$:     DECB     $OCNT      ;;COUNT BY 1
(1) 060626 003347          BGT      2$              ;;BR IF MORE TO DO
(1) 060630 002402          BLT      6$              ;;BR IF DONE
(1) 060632 005204          INC      R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1) 060634 000744          BR       2$              ;;GO DO THE LAST DIGIT
(1) 060636 012605          6$:     MOV      (SP)+,R5      ;;RESTORE R5
(1) 060640 012604          MOV      (SP)+,R4        ;;RESTORE R4
(1) 060642 012603          MOV      (SP)+,R3        ;;RESTORE R3
(1) 060644 016666 000002 000004  MOV      2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
(1) 060652 012616          MOV      (SP)+,(SP)
(1) 060654 000002          RTI
(1) 060656 000          8$:     .BYTE    0          ;;RETURN
(1) 060657 000          .BYTE    0          ;;STORAGE FOR ASCII DIGIT
(1) 060660 000          $OCNT:   .BYTE    0          ;;TERMINATOR FOR TYPE ROUTINE
(1) 060661 000          $OFILL:  .BYTE    0          ;;OCTAL DIGIT COUNTER
(1)                                     .BYTE    0          ;;ZERO FILL SWITCH

```

```

(1) 060662 000000 SOMODE: .WORD 0 ;:NUMBER OF DIGITS TO TYPE
6575 ;*****
(1)
(1) .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1) ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1) ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1) ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1) ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1) ;*REPLACED WITH SPACES.
(1) ;*CALL:
(1) ;* MOV NUM,-(SP) ;:PUT THE BINARY NUMBER ON THE STACK
(1) ;* TYPDS ;:GO TO THE ROUTINE
(1)
(2) 060664 STYPDS:
(3) 060664 010046 MOV R0,-(SP) ;:PUSH R0 ON STACK
(3) 060666 010146 MOV R1,-(SP) ;:PUSH R1 ON STACK
(3) 060670 010246 MOV R2,-(SP) ;:PUSH R2 ON STACK
(3) 060672 010346 MOV R3,-(SP) ;:PUSH R3 ON STACK
(3) 060674 010546 MOV R5,-(SP) ;:PUSH R5 ON STACK
(1) 060676 012746 020200 MOV #20200,-(SP) ;:SET BLANK SWITCH AND SIGN
(1) 060702 016605 000020 MOV 20(SP),R5 ;:GET THE INPUT NUMBER
(1) 060706 100004 BPL 1$ ;:BR IF INPUT IS POS.
(1) 060710 005405 NEG R5 ;:MAKE THE BINARY NUMBER POS.
(1) 060712 112766 000055 000001 MOVB #'-,1(SP) ;:MAKE THE ASCII NUMBER NEG.
(1) 060720 005000 1$: CLR R0 ;:ZERO THE CONSTANTS INDEX
(1) 060722 012703 061100 MOV #SDBLK,R3 ;:SETUP THE OUTPUT POINTER
(1) 060726 112723 000040 MOVB #' ,(R3)+ ;:SET THE FIRST CHARACTER TO A BLANK
(1) 060732 005002 2$: CLR R2 ;:CLEAR THE BCD NUMBER
(1) 060734 016001 061070 MOV $DTBL(R0),R1 ;:GET THE CONSTANT
(1) 060740 160105 3$: SUB R1,R5 ;:FORM THIS BCD DIGIT
(1) 060742 002402 BLT 4$ ;:BR IF DONE
(1) 060744 005202 INC R2 ;:INCREASE THE BCD DIGIT BY 1
(1) 060746 000774 BR 3$
(1) 060750 060105 4$: ADD R1,R5 ;:ADD BACK THE CONSTANT
(1) 060752 005702 TST R2 ;:CHECK IF BCD DIGIT=0
(1) 060754 001002 BNE 5$ ;:FALL THROUGH IF 0
(1) 060756 105716 TSTB (SP) ;:STILL DOING LEADING 0'S?
(1) 060760 100407 BMI 7$ ;:BR IF YES
(1) 060762 106316 5$: ASLB (SP) ;:MSD?
(1) 060764 103003 BCC 6$ ;:BR IF NO
(1) 060766 116663 000001 177777 MOVB 1(SP),-1(R3) ;:YES--SET THE SIGN
(1) 060774 052702 000060 6$: BIS #'0,R2 ;:MAKE THE BCD DIGIT ASCII
(1) 061000 052702 000040 7$: BIS #' ,R2 ;:MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 061004 110223 MOVB R2,(R3)+ ;:PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 061006 005720 TST (R0)+ ;:JUST INCREMENTING
(1) 061010 020027 000010 CMP R0,#10 ;:CHECK THE TABLE INDEX
(1) 061014 002746 BLT 2$ ;:GO DO THE NEXT DIGIT
(1) 061016 003002 BGT 8$ ;:GO TO EXIT
(1) 061020 010502 MOV R5,R2 ;:GET THE LSD
(1) 061022 000764 BR 6$ ;:GO CHANGE TO ASCII
(1) 061024 105726 8$: TSTB (SP)+ ;:WAS THE LSD THE FIRST NON-ZERO?
(1) 061026 100003 BPL 9$ ;:BR IF NO
(1) 061030 116663 177777 177776 MOVB -1(SP),-2(R3) ;:YES--SET THE SIGN FOR TYPING
    
```

```

(1) 061036 105013          9S:  CLRB   (R3)          ;;SET THE TERMINATOR
(3) 061040 012605          MOV   (SP)+,R5        ;;POP STACK INTO R5
(3) 061042 012603          MOV   (SP)+,R3        ;;POP STACK INTO R3
(3) 061044 012602          MOV   (SP)+,R2        ;;POP STACK INTO R2
(3) 061046 012601          MOV   (SP)+,R1        ;;POP STACK INTO R1
(3) 061050 012500          MOV   (SP)+,R0        ;;POP STACK INTO R0
(1) 061052 104400 061100  TYPE   $DBLK          ;;NOW TYPE THE NUMBER
(1) 061056 016666 000002 000004 MOV   2(SP),4(SP)    ;;ADJUST THE STACK
(1) 061064 012616          MOV   (SP)+,(SP)
(1) 061066 000002          RTI                    ;;RETURN TO USER
(1) 061070 023420          $DTBL: 10000.
(1) 061072 001750          1000.
(1) 061074 000144          100.
(1) 061076 000012          10.
(1) 061100 000004          $DBLK: .BLKW 4
6576 ;*****
(1) .SBTTL TTY INPUT ROUTINE
(1) ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*CALL:
(1) ;*   RDCHR          ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*   RETURN HERE   ;;CHARACTER IS ON THE STACK
(1) ;
(1) $RDCHR: MOV   (SP),-(SP)          ;;PUSH DOWN THE PC
(1) 061110 011646          MOV   4(SP),2(SP)    ;;SAVE THE PS
(1) 061112 016666 000004 000002 1S:  TSTB  2$TKS          ;;WAIT FOR
(1) 061120 105777 120012          BPL   1$             ;;A CHARACTER
(1) 061124 100375          MOVB  2$TKB,4(SP)    ;;READ THE TTY
(1) 061126 117766 120006 000004 BIC   #1C<177>,4(SP) ;;GET RID OF JUNK IF ANY
(1) 061134 042766 177600 000004 RTI                    ;;GO BACK TO USER
(1) 061142 000002          ;*****
(1) ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1) ;*CALL:
(1) ;*   RDLIN          ;;INPUT A STRING FROM THE TTY
(1) ;*   RETURN HERE   ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1) ;*                ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1) ;
(1) $RDLIN: MOV   R3, -(SP)          ;;SAVE R3
(1) 061144 010346          CLR   -(SP)          ;;CLEAR THE RUBOUT KEY
(1) 061146 005046          MOV   #$TTYIN,R3    ;;GET ADDRESS
(1) 061150 012703 061405 1S:  CMP   #$TTYIN+7,R3    ;;BUFFER FULL?
(1) 061154 022703 061414 2S:  BLOS  4$             ;;BR IF YES
(1) 061160 101456          RDCHR          ;;GO READ ONE CHARACTER FROM THE TTY
(1) 061162 104412          MOVB  (SP)+,(R3)    ;;GET CHARACTER
(1) 061164 112613          CMPB  #177,(R3)     ;;IS IT A RUBOUT
(1) 061166 122713 000177          BNE   5$             ;;IS IT A RUBOUT
(1) 061172 001022          BNE   5$             ;;BR IF NO
(1) 061174 005716          TST   (SP)          ;;IS THIS THE FIRST RUBOUT?
(1) 061176 001007          BNE   6$             ;;BR IF NO
(1) 061200 112737 000134 061376 MOVB  #' \,9$        ;;TYPE A BACK SLASH
(1) 061206 104400 061376          TYPE  9$
(1) 061212 012716 177777          MOV   #-1,(SP)     ;;SET THE RUBOUT KEY
(1) 061216 005303          6S:  DEC   R3          ;;BACKUP BY ONE

```



```

(1) 061416 016666 000004 000002      MOV      4(SP),2(SP)      ;; INPUT NUMBER
(3) 061424 010046                    MOV      RO,-(SP)        ;; PUSH RO ON STACK
(3) 061426 010146                    MOV      R1,-(SP)        ;; PUSH R1 ON STACK
(3) 061430 010246                    MOV      R2,-(SP)        ;; PUSH R2 ON STACK
(1) 061432 104414                    1$: RDLIN                ;; READ AN ASCII LINE
(1) 061434 012600                    MOV      (SP)+,RO        ;; GET ADDRESS OF 1ST CHARACTER
(1) 061436 010037 061542            MOV      RO,$$          ;; AND SAVE IT
(1) 061442 005001                    CLR      R1              ;; CLEAR DATA WORD
(1) 061444 005002                    CLR      R2
(1) 061446 112046                    2$: MOVB      (RO)+,-(SP) ;; PICKUP THIS CHARACTER
(1) 061450 001420                    BEQ      3$              ;; IF ZERO GET OUT
(1) 061452 122716 000060            CMPB     #'0,(SP)        ;; MAKE SURE THIS CHARACTER
(1) 061456 003026                    BGT      4$              ;; IS AN OCTAL DIGIT
(1) 061460 122716 000067            CMPB     #'7,(SP)
(1) 061464 002423                    BLT      4$
(1) 061466 006301                    ASL      R1                ;; #2
(1) 061470 006102                    ROL      R2
(1) 061472 006301                    ASL      R1                ;; #4
(1) 061474 006102                    ROL      R2
(1) 061476 006301                    ASL      R1                ;; #8
(1) 061478 006102                    ROL      R2
(1) 061500 006102                    ROL      R2
(1) 061502 042716 177770            BIC      #'C7,(SP)        ;; STRIP THE ASCII JUNK
(1) 061506 062601                    ADD      (SP)+,R1        ;; ADD IN THIS DIGIT
(1) 061510 000756                    BR       2$              ;; LOOP
(1) 061512 005726                    3$: TST      (SP)+        ;; CLEAN TERMINATOR FROM STACK
(1) 061514 010166 000012            MOV      R1,12(SP)       ;; SAVE THE RESULT
(1) 061520 010237 061552            MOV      R2,$HIOCT
(3) 061524 012602                    MOV      (SP)+,R2        ;; POP STACK INTO R2
(3) 061526 012601                    MOV      (SP)+,R1        ;; POP STACK INTO R1
(3) 061530 012600                    MOV      (SP)+,RO        ;; POP STACK INTO RO
(1) 061532 000002                    RTI                       ;; RETURN
(1) 061534 005726                    4$: TST      (SP)+        ;; CLEAN PARTIAL FROM STACK
(1) 061536 105010                    CLRB     (RO)            ;; SET A TERMINATOR
(1) 061540 104400                    TYPE     ;; TYPE UP THRU THE BAD CHAR.
(1) 061542 000000                    5$: .WORD     0
(1) 061544 104400 001200            TYPE     $QUES           ;; "?" "CR" & "LF"
(1) 061550 000730                    BR       1$              ;; TRY AGAIN
(1) 061552 000000                    $HIOCT: .WORD     0      ;; HIGH ORDER BITS GO HERE
*****
(1)                                     .SBTTL  SAVE AND RESTORE RO-R5 ROUTINES
(1)
(1)                                     ;; *SAVE RO-R5
(1)                                     ;; *CALL:
(1)                                     ;; * SAVREG
(1)                                     ;; *UPON RETURN FROM $$SAVREG THE STACK WILL LOOK LIKE:
(1)                                     ;; *
(1)                                     ;; *TOP---(+16)
(1)                                     ;; * +2---(+18)
(1)                                     ;; * +4---R5
(1)                                     ;; * +6---R4
(1)                                     ;; * +8---R3
(1)                                     ;; * +10---R2
(1)                                     ;; * +12---R1

```

6578

```

(1) ;#+14---RO
(1)
(1) 061554          ;SSAVREG:
(3) 061554 010046  MOV      RO,-(SP)      ;; PUSH RO ON STACK
(3) 061556 010146  MOV      R1,-(SP)      ;; PUSH R1 ON STACK
(3) 061560 010246  MOV      R2,-(SP)      ;; PUSH R2 ON STACK
(3) 061562 010346  MOV      R3,-(SP)      ;; PUSH R3 ON STACK
(3) 061564 010446  MOV      R4,-(SP)      ;; PUSH R4 ON STACK
(3) 061566 010546  MOV      R5,-(SP)      ;; PUSH R5 ON STACK
(1) 061570 016646 000022  MOV      22(SP),-(SP)  ;; SAVE PS OF MAIN FLOW
(1) 061574 016646 000022  MOV      22(SP),-(SP)  ;; SAVE PC OF MAIN FLOW
(1) 061600 016646 000022  MOV      22(SP),-(SP)  ;; SAVE PS OF CALL
(1) 061604 016646 000022  MOV      22(SP),-(SP)  ;; SAVE PC OF CALL
(1) 061610 000002  RTI

```

```

(1) ;*RESTORE RO-R5
(1) ;*CALL:
(1) ;* RESREG
(1) ;SRESREG:
(1) 061612          MOV      (SP)+,22(SP)  ;; RESTORE PC OF CALL
(1) 061616 012666 000022  MOV      (SP)+,22(SP)  ;; RESTORE PS OF CALL
(1) 061622 012666 000022  MOV      (SP)+,22(SP)  ;; RESTORE PC OF MAIN FLOW
(1) 061626 012666 000022  MOV      (SP)+,22(SP)  ;; RESTORE PS OF MAIN FLOW
(3) 061632 012605  MOV      (SP)+,R5      ;; POP STACK INTO R5
(3) 061634 012604  MOV      (SP)+,R4      ;; POP STACK INTO R4
(3) 061636 012603  MOV      (SP)+,R3      ;; POP STACK INTO R3
(3) 061640 012602  MOV      (SP)+,R2      ;; POP STACK INTO R2
(3) 061642 012601  MOV      (SP)+,R1      ;; POP STACK INTO R1
(3) 061644 012600  MOV      (SP)+,R0      ;; POP STACK INTO R0
(1) 061646 000002  RTI

```

```

6579 ;*****
(1)
(1) .SBTTL TRAP DECODER
(1)
(1) ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1) ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1) ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1) ;*GO TO THAT ROUTINE.

```

```

(1) 061650 010046          STRAP: MOV      RO,-(SP)      ;; SAVE RO
(1) 061652 016600 000002  MOV      2(SP),RO      ;; GET TRAP ADDRESS
(1) 061656 005740          TST      -(RO)          ;; BACKUP BY 2
(1) 061660 111000          MOVB    (RO),RO        ;; GET RIGHT BYTE OF TRAP
(1) 061662 016000 061670  MOV      $TRPAD(RO),RO  ;; INDEX TO TABLE
(1) 061666 000200          RTS      RO            ;; GO TO ROUTINE

```

```

(3)
(3) .SBTTL TRAP TABLE
(3)
(3) ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3) ;*BY THE "TRAP" INSTRUCTION.
(3)
(3) : ROUTINE
(3) : -----

```

(3) 061670  
 (3) 061670 060330  
 (3) 061672 060462  
 (3) 061674 060436  
 (3) 061676 060476  
 (3) 061700 060664  
 (3) 061702 061110  
 (3) 061704 061144  
 (3) 061706 061414  
 (3) 061710 061554  
 (3) 061712 061612

STRPAD:

STYPE ;;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE  
 STYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)  
 STYPOS ;;CALL=TYPOS TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZEROS)  
 STYPON ;;CALL=TYPON TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST CALL)  
 STYPDS ;;CALL=TYPDS TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)  
 SRDCHR ;;CALL=RDCHR TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE  
 SRDLIN ;;CALL=RDLIN TRAP+14(104414) TTY TYPEIN STRING ROUTINE  
 SRDOCT ;;CALL=RDOCT TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY  
 \$SAVREG ;;CALL=SAVREG TRAP+20(104420) SAVE R0-R5 ROUTINE  
 \$RESREG ;;CALL=RESREG TRAP+22(104422) RESTORE R0-R5 ROUTINE

6580  
 6581  
 6582  
 6583  
 6584  
 6585  
 6586  
 6587

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

.SBTTL TELETYPE MESSAGES

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

061714 005015 042115 030455  
 061722 026461 055104 050122  
 061730 026520 006501 005012  
 6588 061736 050122 032060 042040  
 061744 040525 020114 047503  
 061752 052116 047522 046114  
 061760 051105 046040 043517  
 061766 041511 052040 051505  
 061774 020124 020055 040520  
 062002 052122 030440 005015  
 062010 000012  
 6589 062012 005015 047105 042524  
 062020 020122 051104 053111  
 062026 020105 042101 051104  
 062034 051505 035123 000040  
 6590 062042 047111 040526 044514  
 062050 020104 042101 051104  
 062056 051505 006523 000012  
 6591 062064 005015 047520 052122  
 062072 040440 040440 042104  
 062100 042522 051523 044440  
 062106 035123 000040  
 6592 062112 005015 047520 052122  
 062120 041040 040440 042104  
 062126 042522 051523 044440  
 062134 035123 000040  
 6593 062140 005015 054523 052123  
 062146 046505 046440 051525  
 062154 020124 040510 042526  
 062162 023440 023514 047440  
 062170 020122 050047 020047  
 062176 046103 041517 006513  
 062204 005012 000  
 6594 062207 012 047105 042524  
 062214 020122 042524 052123  
 062222 021440 020072 000

TITLE: .ASCII <15><12>/MD-11-DZRPP-A/<15><12><12>

.ASCIIZ /RPO4 DUAL CONTROLLER LOGIC TEST - PART 1/<15><12><12>

ENTERA: .ASCIIZ <15><12>/ENTER DRIVE ADDRESS: /

ADRERR: .ASCIIZ /INVALID ADDRESS/<15><12>

PORTAIS: .ASCIIZ <15><12>/PORT A ADDRESS IS: /

PORTBIS: .ASCIIZ <15><12>/PORT B ADDRESS IS: /

NOCLOCK: .ASCIIZ <15><12>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<15><12><12>

TESTNO: .ASCIIZ <12>/ENTER TEST #: /



6595	062227	111	053116	046101	BADNO: .ASCIZ /INVALID TEST NUMBER/<15><12>
	062234	042111	052040	051505	
	062242	020124	052516	041115	
	062250	051105	005015	000	
6596	062255	040	051105	047522	TSTERR: .ASCIZ / ERRORS/<15><12>
	062262	051522	005015	000	
6597	062267	015	005012	044124	ADDRIS: .ASCIZ <15><12><12>/THE PRESENT ADDRESS OF THE RH11 (RHCS1) IS: /
	062274	020105	051120	051505	
	062302	047105	020124	042101	
	062310	051104	051505	020123	
	062316	043117	052040	042510	
	062324	051040	030510	020061	
	062332	051050	041510	030523	
	062340	020051	051511	020072	
	062346	000			
6598	062347	012	047105	042524	NTRH11: .ASCIZ <12>/ENTER NEW RH11 ADDRESS: /
	062354	020122	042516	020127	
	062362	044122	030461	040440	
	062370	042104	042522	051523	
	062376	020072	000		
6599	062401	015	052012	042510	NORESP: .ASCIZ <15><12>/THE RH11 DID NOT RESPOND WHEN RHWC ACCESSED AT ADDR: /
	062406	051040	030510	020061	
	062414	044504	020104	047516	
	062422	020124	042522	050123	
	062430	047117	020104	044127	
	062436	047105	051040	053510	
	062444	020103	041501	042503	
	062452	051523	042105	040440	
	062460	020124	042101	051104	
	062466	020072	000		
6600					
6601					;;*****
6602					
6603					.SBTTL TEST ERROR MESSAGES
6604					
6605					;;*****
6606					
6607	062471	127	047522	043516	EM1: .ASCIZ /WRONG DRIVE TYPE/
	062476	042040	044522	042526	
	062504	052040	050131	000105	
6608					
6609	062512	051104	053111	020105	EM2: .ASCIZ /DRIVE NOT ON LINE/
	062520	047516	020124	047117	
	062526	046040	047111	000105	
6610					
6611	062534	042523	044522	046101	EM3: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
	062542	047040	046525	042502	
	062550	020122	042522	042101	
	062556	052040	051110	052517	
	062564	044107	042440	041501	
	062572	020110	047520	052122	
	062600	047040	052117	052040	
	062606	042510	051440	046501	
	062614	000105			

6612					
6613	062616	051104	053111	020105	EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/
	062624	047516	020124	042523	
	062632	055111	042105	041040	
	062640	020131	047520	052122	
	062646	000			
6614					
6615	062647	127	047522	043516	EM5: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
	062654	051440	040524	052524	
	062662	020123	042523	047105	
	062670	041040	020131	044124	
	062676	020105	042523	055111	
	062704	047111	020107	047520	
	062712	052122	000		
6616					
6617	062715	122	043505	051511	EM6: .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
	062722	042524	020122	047503	
	062730	052116	047105	051524	
	062736	053440	051105	020105	
	062744	042523	047105	041040	
	062752	020131	050117	047520	
	062760	044523	042524	050040	
	062766	051117	020124	020055	
	062774	051104	053111	020105	
	063002	040527	020123	042523	
	063010	055111	042105	000	
6618					
6619	063015	122	043505	051511	EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
	063022	042524	020122	047503	
	063030	052116	047105	051524	
	063036	053440	047522	043516	
	063044	040440	052106	051105	
	063052	051040	046105	040505	
	063060	042523	047440	020122	
	063066	044524	042515	052517	
	063074	000124			
6620					
6621	063076	042522	044507	052123	EM10: .ASCIZ /REGISTER CONTENTS WRONG/
	063104	051105	041440	047117	
	063112	042524	052116	020123	
	063120	051127	047117	000107	
6622					
6623	063126	047503	052116	047522	EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
	063134	020114	052502	020123	
	063142	040520	044522	054524	
	063150	042440	051122	051117	
	063156	051040	040505	044504	
	063164	043516	044440	042116	
	063172	041511	052101	042105	
	063200	051040	043505	051511	
	063206	042524	000122		
6624					
6625	063212	051104	053111	020105	EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
	063220	047516	020124	042523	

	063226	055111	042105	041040	
	063234	020131	051104	053111	
	063242	020105	046103	040505	
	063250	020122	047503	046515	
	063256	047101	000104		
6626					
6627	063262	042522	042101	047111	EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
	063270	050040	042522	042523	
	063276	020124	047504	051505	
	063304	047040	052117	051440	
	063312	052105	053040	046117	
	063320	046525	020105	040526	
	063326	044514	020104	047506	
	063334	020122	044124	020105	
	063342	047520	052122	000	
6628					
6629	063347	126	046117	046525	EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
	063354	020105	040526	044514	
	063362	020104	042523	020124	
	063370	047117	052040	042510	
	063376	053440	047522	043516	
	063404	050040	051117	000124	
6630					
6631	063412	052101	047124	041040	EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
	063420	052111	053440	047522	
	063426	043516	040440	052106	
	063434	051105	052040	046511	
	063442	047505	052125	026440	
	063450	051040	050505	042525	
	063456	052123	047040	052117	
	063464	051440	052105	000	
6632					
6633	063471	101	052124	020116	EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
	063476	044502	020124	051127	
	063504	047117	020107	043101	
	063512	042524	020122	042522	
	063520	042514	051501	020105	
	063526	020055	042522	052521	
	063534	051505	020124	042523	
	063542	000124			
6634					
6635	063544	052101	047124	041040	EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
	063552	052111	053440	047522	
	063560	043516	040440	052106	
	063566	051105	051040	046105	
	063574	040505	042523	026440	
	063602	051040	050505	042525	
	063610	052123	047040	052117	
	063616	051440	052105	000	
6636					
6637	063623	104	044522	042526	EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
	063630	047040	052117	051440	
	063636	044505	042532	020104	
	063644	044127	047105	040440	

	063652	052124	020116	044502	
	063660	020124	047506	020122	
	063666	047520	052122	041440	
	063674	042514	051101	042105	
	063702	000			
6638					
6639	063703	104	044522	042526	EM21: .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
	063710	051440	044505	042532	
	063716	020104	044127	047105	
	063724	055040	051105	020117	
	063732	051127	052111	042524	
	063740	020116	047111	040440	
	063746	052124	020116	044502	
	063754	000124			
6640					
6641	063756	051104	053111	020105	EM22: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
	063764	047516	020124	047111	
	063772	047040	052505	051124	
	064000	046101	040440	052106	
	064006	051105	052040	046511	
	064014	047505	052125	026440	
	064022	051040	050505	042525	
	064030	052123	047040	052117	
	064036	051440	052105	000	
6642					
6643	064043	124	046511	047505	EM23: .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
	064050	052125	041440	042514	
	064056	051101	042105	052040	
	064064	042510	042040	044522	
	064072	042526	051447	042440	
	064100	051122	051117	041040	
	064106	052111	000		
6644					
6645	064111	122	046105	040505	EM24: .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
	064116	042523	041440	046517	
	064124	040515	042116	051040	
	064132	046105	040505	042523	
	064140	020104	051104	053111	
	064146	020105	044527	044124	
	064154	042440	051122	051117	
	064162	020123	042523	000124	
6646					
6647	064170	044524	042515	052517	EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
	064176	020124	047117	026505	
	064204	044123	052117	042040	
	064212	042111	047040	052117	
	064220	051040	052105	044522	
	064226	043507	051105	000	
6648					
6649	064233	104	044522	042526	EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
	064240	047040	052117	044440	
	064246	020116	042516	052125	
	064254	040522	020114	043101	
	064262	042524	020122	042522	

	064270	042514	051501	020105	
	064276	020055	042522	052521	
	064304	051505	020124	047516	
	064312	020124	042523	000124	
6650					
6651	064320	042522	044507	052123	EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
	064326	051105	053440	047522	
	064334	043516	040440	052106	
	064342	051105	051040	046105	
	064350	040505	042523	053440	
	064356	052111	020110	042522	
	064364	052521	051505	020124	
	064372	042523	000124		
6652					
6653	064376	051104	053111	020105	EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
	064404	042523	055111	042105	
	064412	041040	020131	042522	
	064420	042514	051501	020105	
	064426	047503	046515	047101	
	064434	020104	051511	052523	
	064442	042105	053440	042510	
	064450	020116	051104	053111	
	064456	020105	047111	047040	
	064464	052505	051124	046101	
	064472	000			
6654					
6655	064473	104	044522	042526	EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
	064500	044440	020116	042516	
	064506	052125	040522	020114	
	064514	043101	042524	020122	
	064522	042522	042514	051501	
	064530	020105	020055	042522	
	064536	052521	051505	020124	
	064544	042523	000124		
6656					
6657	064550	052101	047124	041040	EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
	064556	052111	053440	047522	
	064564	043516	040440	052106	
	064572	051105	051040	041505	
	064600	046101	041111	040522	
	064606	042524	041440	046517	
	064614	040515	042116	000	
6658					
6659	064621	104	044522	042526	EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
	064626	051040	052105	051125	
	064634	042516	020104	047524	
	064642	047040	052505	051124	
	064650	046101	044440	020106	
	064656	051104	053111	020105	
	064664	046103	040505	020122	
	064672	044507	042526	020116	
	064700	044127	046111	020105	
	064706	051104	053111	020105	
	064714	042523	055111	042105	

6660	064722	000			
6661	064723	104	044522	042526	EM34: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
	064730	051040	052105	051125	
	064736	042516	020104	047524	
	064744	047040	052505	051124	
	064752	046101	044440	020106	
	064760	040515	051523	052502	
	064766	020123	047111	052111	
	064774	043440	053111	047105	
	065002	053440	044510	042514	
	065010	042040	044522	042526	
	065016	051440	044505	042532	
	065024	000104			
6662					
6663	065026	044524	042515	052517	EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
	065034	020124	047117	020105	
	065042	044123	052117	043040	
	065050	051111	042105	053440	
	065056	052111	047510	052125	
	065064	051040	043505	051511	
	065072	042524	020122	041501	
	065100	042503	051523	000	
6664					
6665	065105	124	046511	047505	EM36: .ASCIZ /TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS/
	065112	052125	044040	051501	
	065120	047040	052117	047440	
	065126	041503	051125	042105	
	065134	053440	052111	044510	
	065142	020116	020062	042523	
	065150	047503	042116	000123	
6666					
6667	065156	051104	053111	020105	EM37: .ASCIZ /DRIVE IS NON-EXISTANT ('NED' BIT SET)/
	065164	051511	047040	047117	
	065172	042455	044530	052123	
	065200	047101	020124	023450	
	065206	042516	023504	041040	
	065214	052111	051440	052105	
	065222	000051			
6668					
6669	065224	052101	047124	041040	EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
	065232	052111	043040	051117	
	065240	050040	051117	020124	
	065246	047516	020124	042522	
	065254	042523	020124	054502	
	065262	046440	051501	041123	
	065270	051525	041440	042514	
	065276	051101	000		
6670					
6671	065301	124	046511	047505	EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
	065306	052125	041440	042514	
	065314	051101	042105	052040	
	065322	042510	040440	052124	
	065330	047105	044524	047117	

	065336	041040	052111	000	
6672	065343	104	044522	042526	EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
6673	065350	047040	052117	044440	
	065356	020116	042516	052125	
	065364	040522	020114	051117	
	065372	051440	044505	042532	
	065400	020104	043101	042524	
	065406	020122	052101	047124	
	065414	041040	052111	053440	
	065422	044522	052124	047105	
	065430	000			
6674	065431	104	044522	042526	EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
6675	065436	044440	020116	042516	
	065444	052125	040522	020114	
	065452	043101	042524	020122	
	065460	052101	042524	052116	
	065466	047511	020116	044502	
	065474	020124	051127	052111	
	065502	042524	000116		
6676	065506	051127	052111	020105	EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
6677	065514	052101	042524	052116	
	065522	047511	020116	044502	
	065530	020124	044504	020104	
	065536	047516	020124	042523	
	065544	020124	047520	052122	
	065552	051040	050505	042525	
	065560	052123	000		
6678	065563	103	047117	051124	EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
6679	065570	046117	042514	020122	
	065576	042523	042514	052103	
	065604	051440	044527	041524	
	065612	020110	047117	042040	
	065620	044522	042526	047040	
	065626	052117	044440	020116	
	065634	040447	041057	000047	
6680	065642	040503	023516	020124	EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
6681	065650	041501	042503	051523	
	065656	042040	044522	042526	
	065664	052040	051110	052517	
	065672	044107	042440	052111	
	065700	042510	020122	047520	
	065706	052122	000		
6682	065711	101	052124	020116	EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
6683	065716	044502	020124	047506	
	065724	020122	042523	055111	
	065732	047111	020107	047520	
	065740	052122	047040	052117	
	065746	041440	042514	051101	

	065754	042105	041040	020131	
	065762	040515	051523	052502	
	065770	020123	047111	052111	
	065776	000			
6684					
6685	065777	101	052124	020116	EMS0: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
	066004	044502	020124	047506	
	066012	020122	050117	047520	
	066020	044523	042524	050040	
	066026	051117	020124	046103	
	066034	040505	042522	020104	
	066042	054502	046440	051501	
	066050	041123	051525	044440	
	066056	044516	000124		
6686					
6687	066062	052101	047124	041040	EMS1: .ASCIZ /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
	066070	052111	041440	042514	
	066076	051101	042105	041040	
	066104	020131	040515	051523	
	066112	052502	020123	047111	
	066120	052111	020054	051104	
	066126	053111	020105	047111	
	066134	047040	052505	051124	
	066142	046101	000		
6688					
6689	066145	124	042510	040440	EMS2: .ASCIZ /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
	066152	052124	020116	044502	
	066160	020124	051511	051440	
	066166	052105	040440	052106	
	066174	051105	052040	046511	
	066202	047505	052125	053440	
	066210	052111	020110	047516	
	066216	051040	050505	042525	
	066224	052123	023040	023440	
	066232	051105	023522	051440	
	066240	052105	000		
6690					
6691	066243	103	047101	052047	EMS3: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
	066250	051040	040505	020104	
	066256	044124	020105	052101	
	066264	047124	041040	052111	
	066272	043040	047522	020115	
	066300	044124	020105	047447	
	066306	050120	051517	052111	
	066314	023505	050040	051117	
	066322	000124			
6692					
6693	066324	042522	042514	051501	EMS4: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
	066332	020105	047503	046515	
	066340	047101	020104	042522	
	066346	047503	047107	055111	
	066354	042105	053440	042510	
	066362	020116	051511	052523	
	066370	042105	041040	020131	



	066376	047516	026516	042523		
	066404	055111	047111	020107		
	066412	047520	052122	000		
6694						
6695	066417	124	046511	047505	EM55:	.ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
	066424	052125	047440	042516		
	066432	051455	047510	020124		
	066440	051511	046040	051505		
	066446	020123	044124	047101		
	066454	032440	030060	046440		
	066462	000123				
6696						
6697						
6698						
6699						
6700	066464	042524	052123	021440	DH1:	.ASCIZ /TEST # ERR PC PORT # REG ADR CONTENTS/
	066472	020040	051105	020122		
	066500	041520	020040	047520		
	066506	052122	021440	020040		
	066514	042522	020107	042101		
	066522	020122	047503	052116		
	066530	047105	051524	000		
6701	066535	124	051505	020124	DH3:	.ASCIZ /TEST # ERR PC REG ADR PORT A PORT B/
	066542	020043	042440	051122		
	066550	050040	020103	051040		
	066556	043505	040440	051104		
	066564	050040	051117	020124		
	066572	020101	050040	051117		
	066600	020124	050102			
6702	066604	020040	020040	020040	DH4:	.ASCII / SEIZE ERROR/<15><12>
	066612	020040	020040	020040		
	066620	020040	020040	042523		
	066626	055111	020105	020040		
	066634	051105	047522	006522		
	066642	012				
6703	066643	124	051505	020124		.ASCIZ /TEST # ERR PC PORT # PORT # REG ADR GOOD BAD/
	066650	020043	042440	051122		
	066656	050040	020103	050040		
	066664	051117	020124	020043		
	066672	050040	051117	020124		
	066700	020043	051040	043505		
	066706	040440	051104	043440		
	066714	047517	020104	020040		
	066722	041040	042101	000		
6704	066727	124	051505	020124	DH5:	.ASCIZ /TEST # ERR PC PORT # REG ADR GOOD BAD/
	066734	020043	042440	051122		
	066742	050040	020103	050040		
	066750	051117	020124	020043		
	066756	051040	043505	040440		
	066764	051104	043440	047517		
	066772	020104	020040	041040		
	067000	042101	000			
6705	067003	040	020040	020040	DH7:	.ASCII / RELSNG ERROR/<15><12>
	067010	020040	020040	020040		



	067466	052122	021440	020040					
	067474	042522	020107	042101					
	067502	020122	047503	052116					
	067510	047105	051524	000					
6714	067515	040	020040	020040	DH26:	.ASCII /		RELSNG/<15><12>	
	067522	020040	020040	020040					
	067530	020040	020040	051040					
	067536	046105	047123	006507					
	067544	012							
6715	067545	124	051505	020124		.ASCIZ /TEST # ERR PC PORT #/			
	067552	020043	042440	051122					
	067560	050040	020103	050040					
6716	067566	051117	020124	000043	DH31:	.ASCII /		RELSNG RQSTNG/<15><12>	
	067574	020040	020040	020040					
	067602	020040	020040	020040					
	067610	020040	020040	042522					
	067616	051514	043516	020040					
	067624	050522	052123	043516					
	067632	005015							
6717	067634	042524	052123	021440		.ASCIZ /TEST # ERR PC PORT # PORT #/			
	067642	020040	051105	020122					
	067650	041520	020040	047520					
	067656	052122	021440	020040					
	067664	047520	052122	021440					
	067672	000							
6718	067673	124	051505	020124	DH36:	.ASCIZ /TEST # ERR PC PORT #/			
	067700	020043	042440	051122					
	067706	050040	020103	050040					
6719	067714	051117	020124	000043	DH42:	.ASCIZ /TEST # ERR PC/			
	067722	042524	052123	021440					
	067730	020040	051105	020122					
	067736	041520	000						
6720	067741	040	020040	020040	DH44:	.ASCII /		RELSNG ERROR/<15><12>	
	067746	020040	020040	020040					
	067754	020040	020040	051040					
	067762	046105	047123	020107					
	067770	042440	051122	051117					
	067776	005015							
6721	070000	042524	052123	021440		.ASCIZ /TEST # ERR PC PORT # PORT #/			
	070006	020040	051105	020122					
	070014	041520	020040	047520					
	070022	052122	021440	020040					
	070030	047520	052122	021440					
	070036	000							
6722	070037	040	020040	020040	DH46:	.ASCII /		PORT A PORT B/<15><12>	
	070044	020040	020040	020040					
	070052	020040	020040	050040					
	070060	051117	020124	020101					
	070066	050040	051117	020124					
	070074	006502	012						
6723	070077	124	051505	020124		.ASCIZ /TEST # ERR PC RHDS1 RHDS1/			
	070104	020043	042440	051122					
	070112	050040	020103	051040					
	070120	042110	030523	020040					

	070126	051040	042110	030523					
	070134	000							
6724	070135	124	051505	020124	DH55:	.ASCIZ	/TEST #	ERR PC	PORT #
	070142	020043	042440	051122					TIMEOUT VALUE (IN MS)/
	070150	050040	020103	050040					
	070156	051117	020124	020043					
	070164	052040	046511	047505					
	070172	052125	053040	046101					
	070200	042525	024040	047111					
	070206	046440	024523	000					
6725									
6726		070214							.EVEN
6727									
6728	070214	001234	001116	001226	DT1:	.WORD	TSTNUM,	SERRPC,	PTNBR,
	070222	001122	001126	000000			\$BDADR,	\$BDDAT,	0
6729	070230	001234	001116	001122	DT3:	.WORD	TSTNUM,	SERRPC,	\$BDADR,
	070236	001124	001126	000000			\$GDDAT,	\$BDDAT,	0
6730	070244	001234	001116	001226	DT5:	.WORD	TSTNUM,	SERRPC,	PTNBR,
	070252	001122	001124	001126			\$BDADR,	\$GDDAT,	\$BDDAT,
	070260	000000					0		
6731	070262	001234	001116	001232	DT6:	.WORD	TSTNUM,	SERRPC,	OPPRT,
	070270	001122	001126	000700			\$BDADR,	\$BDDAT,	0
6732	070276	001234	001116	001230	DT7:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070304	001226	001122	001124			PTNBR,	\$BDADR,	\$GDDAT,
	070312	001126	000000				\$BDDAT,	0	
6733	070316	001234	001116	001230	DT13:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070324	001226	001122	001126			PTNBR,	\$BDADR,	\$BDDAT,
	070332	000000					0		
6734	070334	001234	001116	001230	DT22:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070342	001226	000000				PTNBR,	0	
6735	070346	001234	001116	001230	DT23:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070354	001122	001126	000000			\$BDADR,	\$BDDAT,	0
6736	070362	001234	001116	001230	DT31:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070370	001232	000000				OPPRT,	0	
6737	070374	001234	001116	001230	DT36:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070402	000000					0		
6738	070404	001234	001116	001226	DT37:	.WORD	TSTNUM,	SERRPC,	PTNBR,
	070412	000000					0		
6739	070414	001234	001116	000000	DT42:	.WORD	TSTNUM,	SERRPC,	0
6740	070422	001234	001116	001162	DT46:	.WORD	TSTNUM,	SERRPC,	\$TMP2,
	070430	001164	000000				\$TMP3,	0	
6741	070434	001234	001116	001232	DT54:	.WORD	TSTNUM,	SERRPC,	OPPRT,
	070442	001230	000000				SEIZPT,	0	
6742	070446	001234	001116	001230	DT55:	.WORD	TSTNUM,	SERRPC,	SEIZPT,
	070454	001244	000000				TIME,	0	
6743									
6744	070460	000	000	000	DF1:	.BYTE	0,0,0,0,0		
	070463	000	000						
6745	070465	000	000	000	DF5:	.BYTE	0,0,0,0,0,0		
	070470	000	000	000					
6746	070473	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0		
	070476	000	000	C00					
	070501	000							
6747	070502	000	000	000	DF31:	.BYTE	0,0,0,0		

```

6748 070505 000
6749 070506 000 000 000 DF36: .BYTE 0,0,0
6750 070511 000 000 000 DF42: .BYTE 0,0
6750 070513 000 000 000 DF55: .BYTE 0,0,0,1
6750 070516 001

```

```

6751
6752 070520 .EVEN
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762

```

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

.SBTTL CONSTANTS, TABLES, ETC

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

;TABLE OF TEST STARTING ADDRESSES

```

6763 070520 002712 TSTADR: .WORD TST1+2 ;STARTING ADDRESS OF TEST 1
6766 070522 004244 .WORD TST2+2 ;STARTING ADDRESS OF TEST 2
(1) 070524 005670 .WORD TST3+2 ;STARTING ADDRESS OF TEST 3
(1) 070526 007314 .WORD TST4+2 ;STARTING ADDRESS OF TEST 4
(1) 070530 010500 .WORD TST5+2 ;STARTING ADDRESS OF TEST 5
(1) 070532 011664 .WORD TST6+2 ;STARTING ADDRESS OF TEST 6
(1) 070534 012702 .WORD TST7+2 ;STARTING ADDRESS OF TEST 7
(1) 070536 013720 .WORD TST10+2 ;STARTING ADDRESS OF TEST 10
(1) 070540 014370 .WORD TST11+2 ;STARTING ADDRESS OF TEST 11
(1) 070542 015040 .WORD TST12+2 ;STARTING ADDRESS OF TEST 12
(1) 070544 016442 .WORD TST13+2 ;STARTING ADDRESS OF TEST 13
(1) 070546 020044 .WORD TST14+2 ;STARTING ADDRESS OF TEST 14
(1) 070550 021374 .WORD TST15+2 ;STARTING ADDRESS OF TEST 15
(1) 070552 022724 .WORD TST16+2 ;STARTING ADDRESS OF TEST 16
(1) 070554 024212 .WORD TST17+2 ;STARTING ADDRESS OF TEST 17
(1) 070556 025124 .WORD TST20+2 ;STARTING ADDRESS OF TEST 20
(1) 070560 026036 .WORD TST21+2 ;STARTING ADDRESS OF TEST 21
(1) 070562 027242 .WORD TST22+2 ;STARTING ADDRESS OF TEST 22
(1) 070564 030446 .WORD TST23+2 ;STARTING ADDRESS OF TEST 23
(1) 070566 031502 .WORD TST24+2 ;STARTING ADDRESS OF TEST 24
(1) 070570 032536 .WORD TST25+2 ;STARTING ADDRESS OF TEST 25
6769 070572 032536 .WORD TST25+2 ;STARTING ADDRESS OF TEST 25
(1) 070574 033752 .WORD TST26+2 ;STARTING ADDRESS OF TEST 26
(1) 070576 035166 .WORD TST27+2 ;STARTING ADDRESS OF TEST 27
(1) 070600 037204 .WORD TST30+2 ;STARTING ADDRESS OF TEST 30
(1) 070602 037726 .WORD TST31+2 ;STARTING ADDRESS OF TEST 31
(1) 070604 041214 .WORD TST32+2 ;STARTING ADDRESS OF TEST 32
(1) 070606 042502 .WORD TST33+2 ;STARTING ADDRESS OF TEST 33
(1) 070610 043620 .WORD TST34+2 ;STARTING ADDRESS OF TEST 34
(1) 070612 044736 .WORD TST35+2 ;STARTING ADDRESS OF TEST 35
(1) 070614 045600 .WORD TST36+2 ;STARTING ADDRESS OF TEST 36
6772 070616 046442 .WORD TST37+2 ;STARTING ADDRESS OF TEST 37
(1) 070620 047436 .WORD TST40+2 ;STARTING ADDRESS OF TEST 40
(1) 070622 050432 .WORD TST41+2 ;STARTING ADDRESS OF TEST 41
(1) 070624 051442 .WORD TST42+2 ;STARTING ADDRESS OF TEST 42
(1) 070626 052452 .WORD TST43+2 ;STARTING ADDRESS OF TEST 43
(1) 070630 054040 .WORD TST44+2 ;STARTING ADDRESS OF TEST 44

```

```

(1) 070632 055426            .WORD   TST45+2            ; STARTING ADDRESS OF TEST 45
(1) 070634 056344            .WORD   TST46+2            ; STARTING ADDRESS OF TEST 46
6773
6774                            ; ATTENTION BIT TABLE
6775                            ATABIT: .BYTE   1            ; ATTENTION BIT FOR DRIVE 0
6776 070636            001            .BYTE   2            ; ATTENTION BIT FOR DRIVE 1
6777 070637            002            .BYTE   4            ; ATTENTION BIT FOR DRIVE 2
6778 070640            004            .BYTE  10            ; ATTENTION BIT FOR DRIVE 3
6779 070641            010            .BYTE  20            ; ATTENTION BIT FOR DRIVE 4
6780 070642            020            .BYTE  40            ; ATTENTION BIT FOR DRIVE 5
6781 070643            040            .BYTE 100            ; ATTENTION BIT FOR DRIVE 6
6782 070644            100            .BYTE 200            ; ATTENTION BIT FOR DRIVE 7
6783 070645            200
6784
6785 070646            000046            MAXTN: .WORD   $TN-1            ; MAXIMUM TEST NUMBER
6786
6787                            .END

```



DZRPPA.P11

CROSS REFERENCE TABLE

CKERR	001236	4521#	4984*	4992*	5001*	5202*	5203*	5222*	5226*	5272*	5287*	5387*	5409*	5472*
		5493*	5550*	5648*	5670*	5709*	5728*	5777*	5799*	5872*	5980*	5997*	6058*	6081*
		6233*	6257*	6310*	6329*	6394*	6423*							
CLOCK	057416	6513	6522	6533#										
CLR	= 000040	4269#	4984	5387	5409	5472	5493	5550	6058	6081				
CSF	= 000002	4391#												
CSU	= 000010	4393#												
DCK	= 100000	4340#												
DCL	= 000100	4435#												
DE1	= 000040	4311#												
DFF20	= 000002	4307#												
DF1	070460	4528	4535	4542	4584	4655	4662	4748	4755	4783	4797	4825	6744#	
DF31	070502	4648	4678	4699	4776	4790	4832	6747#						
DF36	070506	4591	4633	4641	4670	4692	4713	4720	4727	4734	4741	6749#		
DF42	070511	4762	4769	6749#										
DF5	070465	4556	4563	4577	4598	4605	4706	4804	4811	4818	6745#			
DF55	070513	4839	6750#											
DF7	070473	4549	4570	4612	4619	4626	4685	6746#						
DH1	066464	4526	4533	4746	4781	6700#								
DH11	067126	4582	6707#											
DH13	067177	4561	4596	4603	4302	4816	6708#							
DH22	067317	4646	4676	4830	6710#									
DH23	067415	4653	4660	4753	4795	4823	6712#							
DH26	067515	6714#												
DH3	066535	4540	6701#											
DH31	067574	4697	6716#											
DH36	067673	4589	4631	4638	4668	4690	4711	4718	4725	4732	4739	6718#		
DH4	066604	4547	6702#											
DH42	067722	4760	4767	6719#										
DH44	067741	4774	6720#											
DH46	070037	4788	6722#											
DH5	066727	4554	4575	4704	4809	6704#								
DH55	070135	4837	6724#											
DH7	067003	4568	4610	4617	4624	4683	6705#							
DIG8	= 000004	4308#												
DISPLA	= 177570	4248#	6570*	6571*										
DLT	= 100000	4279#												
DL64	= 000020	4310#												
DMD	= 000001	4344#												
DPR	= 000400	4314#	5001	5110	5130	5202	5203	5222	5226	5272	5287	5316	5327	5387
		5409	5472	5493	5577	5593	5648	5670	5709	5728	5777	5799	5872	5906
		6058	6081	6141	6160	6233	6257	6310	6329	6393	6422	6480	6496	5387
DRY	= 000200	4313#	5001	5110	5130	5202	5203	5222	5226	5272	5287	5316	5327	5387
		5409	5472	5493	5577	5593	5648	5670	5709	5728	5777	5799	5872	5906
		6058	6081	6141	6160	6233	6257	6310	6329	6393	6422	6480	6496	
DTE	= 010000	4337#												
DTSY	= 000200	4350#												
DT1	070214	4527	4534	4583	4782	4824	6728#							
DT13	070316	4562	4597	4604	4803	4817	6733#							
DT22	070334	4647	4677	6734#										
DT23	070346	4654	4661	4747	4754	4796	6735#							
DT3	070230	4541	6729#											
DT31	070362	4698	4775	6736#										
DT36	070374	4669	4691	4712	4719	4733	6737#							



DT37	070404	4590	4632	4639	4726	4740	6738#
DT42	070414	4761	4768	6739#			
DT46	070422	4789	6740#				
DT5	070244	4555	4576	4705	4810	6730#	
DT54	070434	4831	6741#				
DT55	070446	4838	6742#				
DT6	070262	6731#					
DT7	070276	4548	4569	4611	4618	4625	4684 6732#
DVA	= 004000	4299#					
ECH	= 000100	4331#					
ECI	= 004000	4416#					
EMTVEC	= 000030	4248#	4855*				
EM1	062471	4525	6607#				
EM10	063076	4574	6621#				
EM11	063126	4581	6623#				
EM12	063212	4588	6625#				
EM13	063262	4595	6627#				
EM14	063347	4602	6629#				
EM15	063412	4609	6631#				
EM16	063471	4616	6633#				
EM17	063544	4623	6635#				
EM2	062512	4532	6609#				
EM20	063623	4630	6637#				
EM21	063703	4637	6639#				
EM22	063756	4645	6641#				
EM23	064043	4652	6643#				
EM24	064111	4659	6645#				
EM25	064170	4667	6647#				
EM26	064233	4675	6649#				
EM27	064320	4682	6651#				
EM3	062534	4539	6611#				
EM30	064376	4689	6653#				
EM31	064473	4696	6655#				
EM32	064550	4703	6657#				
EM33	064621	4710	6659#				
EM34	064723	4717	6661#				
EM35	065026	4724	6663#				
EM36	065105	4731	6665#				
EM37	065156	4738	6667#				
EM4	062616	4546	6613#				
EM40	065224	4745	6669#				
EM41	065301	4752	6671#				
EM42	065343	4759	6673#				
EM43	065431	4766	6675#				
EM44	065506	4773	6677#				
EM45	065563	4780	6679#				
EM46	065642	4787	6681#				
EM47	065711	4794	6683#				
EM5	062647	4553	6615#				
EM50	065777	4801	6685#				
EM51	066062	4808	6687#				
EM52	066145	4815	6689#				
EM53	066243	4822	6691#				
EM54	066324	4829	6693#				





PTNBR 001226

4521*	4984*	4992*	5001*	5110*	5130*	5202*	5203*	5222*	5226*	5272*	5287*	5316*
5327*	5387*	5409*	5472*	5493*	5550*	5577*	5593*	5648*	5670*	5709*	5728*	5777*
5799*	5872*	5906*	5980*	5997*	6058*	6081*	6141*	6160*	6233*	6257*	6310*	6329*
6393*	6422*	6480*	6496*	6728	6730	6732	6733	6734	6738			

PWRVEC= 000024  
 RDCHR = 104412  
 RDLIN = 104414  
 RDOCT = 104416  
 RDY = 000200  
 RELERR 001242

RELOK = 000001

4248*	6576	6579*										
6577	6579*											
4861	4906	4934	6579*									
4295*												
4521*	5110*	5130*	5203*	5226*	5272*	5287*	5316*	5327*	5387*	5409*	5472*	5493*
5550*	5577*	5593*	5648*	5670*	5709*	5728*	5777*	5799*	5872*	5906*	5980*	5997*
6058*	6081*	6141*	6160*	6233*	6257*	6310*	6329*	6393*	6422*	6480*	6496*	
4846*	5110	5130	5203	5226	5272	5287	5316	5327	5330*	5387	5409	5472
5493	5550	5577	5593	5648	5670	5709	5728	5777	5799	5872	5906	5980
5997	6058	6081	6141	6160	6233	6257	6310	6329	6393	6422	6480	6496

RESREG= 104422  
 RESVEC= 000010  
 RHAS = 000016  
 RHBA = 000004  
 RHCA = 000034  
 RHCC = 000036  
 RHCS1 = 000000

RHCS2 = 000010

6579*												
4248*												
4458*	5872*	5906*	6393*	6422*	6480	6496						
4453*												
4465*	5110	5130										
4466*	5110	5130										
4451*	5202*	5222*	5272*	5287*	5316*	5327*	5387*	5409*	5472*	5493*	5550*	5577*
5593*	5648*	5670*	5709*	5728*	5777*	5799*	5872*	5906*	5980*	5997*	6058*	6081*
6141*	6160*	6233*	6257*	6310*	6329*	6393*	6394*	6422*	6423*	6480*	6496*	
4455*	4984*	4992*	5001*	5006*	5008*	5110*	5130*	5202*	5203*	5222*	5226*	5272*
5287*	5316*	5327*	5387*	5409*	5472*	5493*	5550*	5577*	5593*	5648*	5670*	5709*
5728*	5777*	5799*	5872*	5906*	5980*	5997*	6058*	6081*	6141*	6160*	6233*	6257*
6310*	6329*	6393*	6422*	6480*	6496*							

RHDA = 000006  
 RHDB = 000022  
 RHDS1 = 000012

4454*	5110	5130										
4460*												
4456*	4984	5001	5110*	5130*	5202	5203	5222	5226	5272*	5287*	5316	5327
5387*	5409*	5472*	5493*	5550	5577*	5593*	5648*	5670*	5709*	5728*	5777*	5799*
5872*	5906*	5980*	5997*	6058*	6081*	6141*	6160*	6233*	6257*	6310*	6329*	6393*
6422*	6480*	6496*										
4462*	4992	5110	5130									
4469*	5110	5130										
4470*	5110	5130										

RHDT = 000026  
 RHEC1 = 000044  
 RHEC2 = 000046  
 RHER1 = 000014

4457*	5110	5130	5472*	5493*	5550*	5980*	5997*	6058*	6081*	6393*	6394	6422*
6423	6480*	6496*										
4467*	5110	5130	6393*	6394	6422*	6423						
4468*	5110	5130	6393*	6394	6422*	6423						
4459*	5110	5130										
4461*	5110	5130										
4464*	5110	5130										
4463*	5007	5009	5202*	5222*								
4452*			5110	5130								
4327*												
4248*												

RHER2 = 000040  
 RHER3 = 000042  
 RHLA = 000020  
 RHMR = 000024  
 RHOF = 000032  
 RHSN = 000030  
 RHWC = 000002  
 RMR = 000004  
 RP6 = 000300  
 RO = %:000000

4248*	4896*	4939*	4940*	4941	4944	4969*	4984*	4992*	5001*	5006*	5007	5008*
5009	5110*	5130*	5202*	5203*	5222*	5226*	5272*	5287*	5316*	5327*	5387*	5409*
5472*	5493*	5550*	5577*	5593*	5648*	5670*	5709*	5728*	5777*	5799*	5872*	5906*
5980*	5997*	6058*	6081*	6141*	6160*	6233*	6257*	6310*	6329*	6393*	6394*	6422*
6423*	6480*	6496*	6566*	6572*	6573*	6575*	6577*	6578*	6579*			
4248*	4883*	4884	4907*	4910	4914*	4915*	4916	6512*	6513*	6514*	6521*	6522*

R1 = %:000001







