RP04

LOGIC TEST PART 1

MD-11-DERPP-A

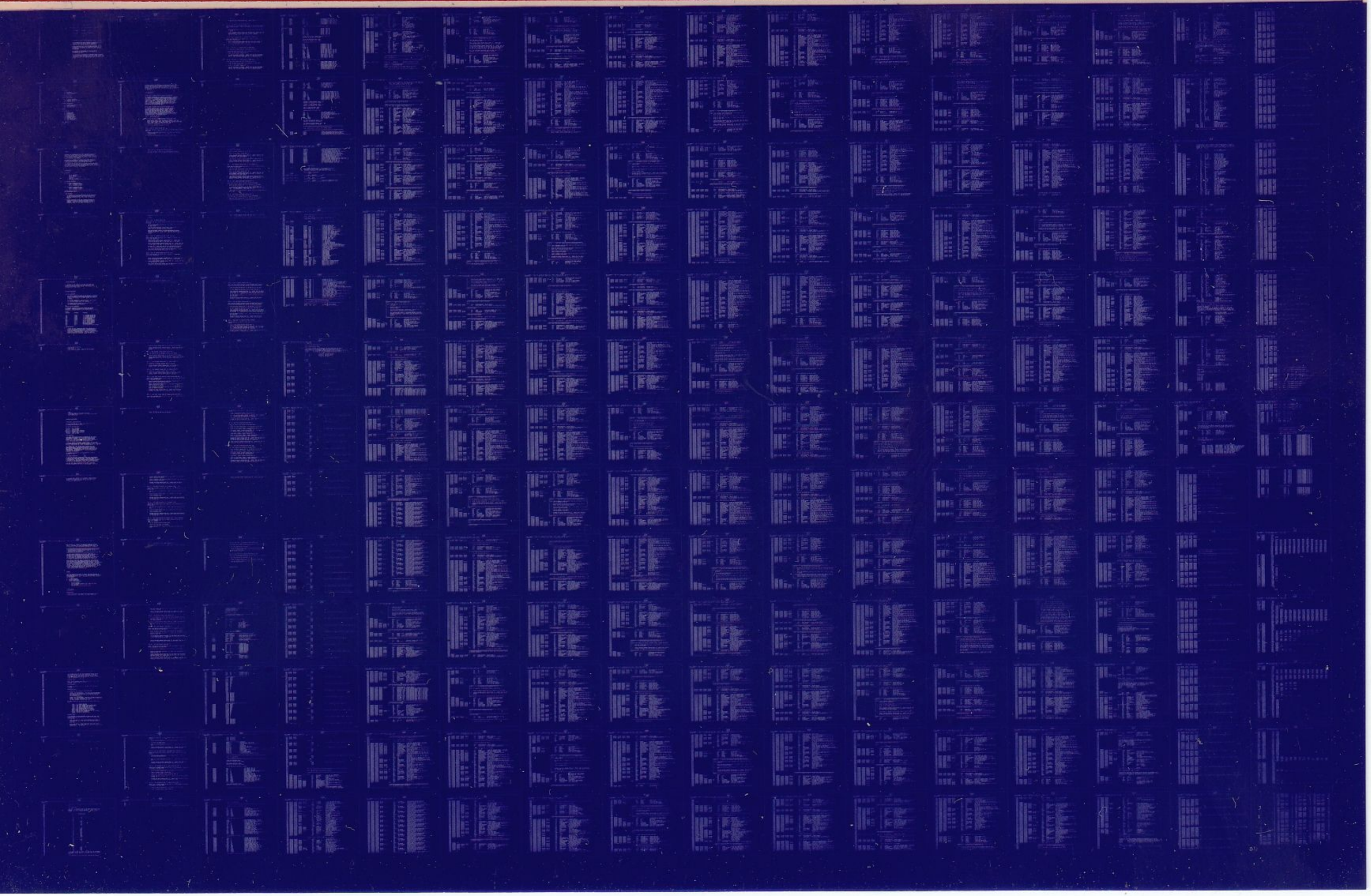
EP-DERPP-A-DL-A

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA



RP04

LOGIC TEST PART 2

MD-11-DERPP-A

EP-DERPP-A-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA

```
                .REM    @
```

I D E N T I F I C A T I O N

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

PRODUCT NAME: RP04 DUAL CONTROLLER LOGIC TEST - PART 1

DATE CREATED: DECEMBER 21, 1974

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: C. HESS

CONTENTS

1. ABSTRACT

THE RP04 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RP04 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 RP04 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/RH70.

THIS PROGRAM IS THE FIRST PART OF THE DUAL CONTROLLER OPTION LOGIC TEST. ALL OF THE DUAL CONTROLLER OPTION LOGIC, EXCEPT THE LOGIC ASSICIATED 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
RH70 WITH AN RP04
RP04 DUAL CONTROLLER OPTION TEST CABLE

2.2 PRELIMINARY PROGRAMS

RP04 DISKLESS CONTROLLER TEST
PART 1 (MAINDEC-11-DERPS)
PART 2 (MAINDEC-11-DERPT)

RP04 FUNCTIONAL CONTROLLER TEST
PART 1 (MAINDEC-11-DERPU)
PART 2 (MAINDEC-11-DERPV)

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

2.3 OTHER PROGRAMS

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

B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RP04 MULTIDRIVE EXERCISER (MAINDEC-11-DERPN-B). NOTE THAT THE RP04 EXERCISER MUST BE PROGRAM REVISION 'B' OR LATER. REVISION 'A' OF THE

```
                                        RP04 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 APPROPIATE MEDIA USING THE ASSOCIATED 'XXDP' LOADED. THE PROGRAM MAY BE INCLUDED IN AN 'XXDP' CHAIN.

4. STARTING PROCEDURES

4.1 STARTING ADDRESSES

A. THE NORMAL STARTING ADDRESS OF THE PROGRAM IS LOCATION 200(8). STARTING AT THIS ADDRESS ALLOWS THE OPERATOR TO SELECT (OR RESELECT) THE 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 RH70 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.

MEMORY LOCATION	CONTENTS	FUNCTION
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	KW11-P STATUS REG
1206	172542	KW11-L COUNTER BUFFER
1210	104	KW11-P VECTOR ADDRESS
1212	177546	KW11-L STATUS REGISTER
1214	100	KW11-L VECTOR ADDRESS
1270	176700	RH70/RP04 ADDRESS
1272	254	RH70 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 APPROPIATE 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 RP04 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 RP04 BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE TEST CABLE CONNECTED TO THE RP04 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT

```
                                        OF THE RP04 WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS.
                                        THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S
                                        ADDRESS (THE ADDRESS SELECTED BY THE SWITCHES ON THE
                                        'DP' BOARD - MODULE M7775.)
```

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/RH7O 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 APPROPIATE 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 TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

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

7.6 REQUIRED TESTS

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

A. TEST 2 AND TEST 3. THESE TESTS DETERMINE AND STORE FOR LATER USE THE TIMEOUT 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.

```
382
383                                 7.7     DISK SURFACE USAGE
384
385                                         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
27	4000
30	4000
31	4
32	4
33	4000
34	4000
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 OCCURING.
BECAUSE THE PROGRAM MUST RESET THE RP04 TO A KNOWN STATE
BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED
AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR
WAS DETECTED

8. TEST DESCRIPTION

8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

THE PROGRAM DETERMINES IF AN RP04 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 VERIFING 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 RP04, 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.

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

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

A. WRITE 0'S INTO 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.

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

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

A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE WAS SEIZIED BY PORT 'A' AND THAT THE 'GO' BIT RESET.

B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT 'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)

C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.

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

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

A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE WAS SEIZIED 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.

```
                                        9.8     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.

8.9 TEST 7 - TEST RELEASE, DRIVE SEIZED BY PORT 'B'

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

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

8.10 TEST 10 - TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

8.11 TEST 11 - TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

8.12 TEST 12 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'

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

A. SEIZE THE DRIVE BY WRITING 0'S INTO 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. SSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.

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

8.13 TEST 13 - TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'

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

```
622
623                                     A.  SEIZE THE DRIVE BY WRITING 0'S INTO RHDS1 THROUGH PORT 'B'.
624                                         VERIFY THAT THE DRIVE HAS BEEN SEIZED.
625
```

B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.

C. ISSUE A MASSBUS CLEAR THROUGH THE RH70 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.

D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

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

A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.

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

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

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

A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.

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

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

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

A. SET THE ATTENTION BITS FOR BOTH PORTS.

B. VERIFY THAT THE DRIVE IS IN NEUTRAL.

C. ISSUE A MASSBUS INIT. VERIFY THAT NEITHER ATTENTION BIT HAS RESET.

```
682
683                                 8.17    TEST 17 - TEST SEIZE BY RHCS1 READ THROUGH PORT 'A'
684
685                                         VERIFY THAT READING THE CONTROL REGISTER (RHCS1) SEIZES THE DRIVE.
```

A. READ THE CONTROL REGISTER (RHCS1) 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 RHCS1 READ THROUGH PORT 'B'

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

A. READ THE CONTROL REGISTER (RHCS1) 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.

```
742
743
744                                8.21    TEST 23 - TEST NO 'PORT REQUEST' WHEN READ RHCS1 THROUGH PORT 'A'
745
```

VERIFY THAT READING THE CONTROL REGISTER (RHCS1) DOES NOT SET 'PORT REQUEST'.

A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RHCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT SET.

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

8.22 TEST 24 - TEST NO 'PORT REQUEST' WHEN READ RHCS1 THROUGH PORT 'B'

VERIFY THAT READING THE CONTROL REGISTER (RHCS1) DOES NOT SET 'PORT REQUEST'.

A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RHCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT SET.

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

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

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

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.

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.

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

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

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.

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

```
802
803                                     C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
804
805                                     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.

```
862
863                                 8.29    TEST 33 - TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
864                                         VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
865
```

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 WITHN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)

D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

8.32 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'.

```
922                                     C.  VERIFY THAT THE TIMEOUT OCCURS WITHN + OR - 25% OF THE SPECIFIED
923                                         TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
924
925                                     D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
```

BIT IS SET.

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.

```
982
983
984                                8.36    TEST 42 - TEST PORT 'B' ATTENTION AFTER A COMMAND
985
```

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.

```
1042
1043                    G.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
1044                        RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
1045
```

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

@

@

```
                                .TITLE  MD-11-DERPP-A, RP04 DUAL CONTROLLER LOGIC TEST - PART 1
                                ;*COPYRIGHT (C) 1974
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY C. HESS
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A2).
                                ;*


                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR


                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570          ;;SWITCH REGISTER
177570                          DISPLAY=SWR

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

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
```

```
1134            000200                  PR4=    200             ;;PRIORITY LEVEL 4
1135            000240                  PR5=    240             ;;PRIORITY LEVEL 5
1136            000300                  RP6=    300             ;;PRIORITY LEVEL 6
1137            000340                  PR7=    340             ;;PRIORITY LEVEL 7
1138
1139                                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1140            100000                  SW15=   100000
1141            040000                  SW14=   40000
1142            020000                  SW13=   20000
1143            010000                  SW12=   10000
1144            004000                  SW11=   4000
1145            002000                  SW10=   2000
1146            001000                  SW09=   1000
1147            000400                  SW08=   400
1148            000200                  SW07=   200
1149            000100                  SW06=   100
1150            000040                  SW05=   40
1151            000020                  SW04=   20
1152            000010                  SW03=   10
1153            000004                  SW02=   4
1154            000002                  SW01=   2
1155            000001                  SW00=   1
1156                                    .EQUIV  SW09,SW9
1157                                    .EQUIV  SW08,SW8
1158                                    .EQUIV  SW07,SW7
1159                                    .EQUIV  SW06,SW6
1160                                    .EQUIV  SW05,SW5
1161                                    .EQUIV  SW04,SW4
1162                                    .EQUIV  SW03,SW3
1163                                    .EQUIV  SW02,SW2
1164                                    .EQUIV  SW01,SW1
1165                                    .EQUIV  SW00,SW0
1166
1167                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1168            100000                  BIT15=  100000
1169            040000                  BIT14=  40000
1170            020000                  BIT13=  20000
1171            010000                  BIT12=  10000
1172            004000                  BIT11=  4000
1173            002000                  BIT10=  2000
1174            001000                  BIT09=  1000
1175            000400                  BIT08=  400
1176            000200                  BIT07=  200
1177            000100                  BIT06=  100
1178            000040                  BIT05=  40
1179            000020                  BIT04=  20
1180            000010                  BIT03=  10
1181            000004                  BIT02=  4
1182            000002                  BIT01=  2
1183            000001                  BIT00=  1
1184                                    .EQUIV  BIT09,BIT9
1185                                    .EQUIV  BIT08,BIT8
1186                                    .EQUIV  BIT07,BIT7
1187                                    .EQUIV  BIT06,BIT6
1188                                    .EQUIV  BIT05,BIT5
1189                                    .EQUIV  BIT04,BIT4
```

```
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

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

                                ;;*************************************************************************

                                .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)

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

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


                                ;;*************************************************************************
```

```
                                .SBTTL  RP04 REGISTERS

                                ;;*****************************************************************************

                                ;CONTROL AND STATUS 1 REGISTER. (#00)

000001                          GO=     1                       ;GO (BIT #0)
000100                          IE=     100                     ;INTERRUPT ENABLE (BIT #6)
000200                          RDY=    200                     ;READY (BIT #7)
000400                          A16=    400                     ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                          A17=    1000                    ;HIGH ORDER UNIBUS BITS (BIT #9)
002000                          PSEL=   2000                    ;PORT SELECT (BIT #10)
004000                          DVA=    4000                    ;DEVICE AVAILABLE (BIT #11)
020000                          MCPE=   20000                   ;MASSBUSS PARITY ERROR (BIT #13)
040000                          TRE=    40000                   ;TRANSFER ERROR (BIT #14)
100000                          SC=     100000                  ;SPECIAL CONDITION (BIT #15)

                                ;STATUS REGISTER (RHDS1) (#01)

                                ;DF5=   1                       DRIVE FORWARD 5"/SEC.  (BIT #0)
000002                          DFF20=  2                       ;DRIVE FORWARD 20"/SEC.  (BIT #1)
000004                          DIGB=   4                       ;DRIVE TO INNER GUARD BAND (BIT #2)
000010                          GRV=    10                      ;GO REVERSE (BIT #3)
000020                          DL64=   20                      ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                          DE1=    40                      ;DIFFERENCE EQUALS 1 (BIT #5)
000100                          VV=     100                     ;VOLUME VALID (BIT #6)
000200                          DRY=    200                     ;DRIVE READY (BIT #7)
000400                          DPR=    400                     ;DRIVE PRESENT (BIT #8)
001000                          PGM=    1000                    ;PROGRAMABLE (BIT #9)
002000                          LST=    2000                    ;LAST SECTOR TRANSFERRED (BIT #10)
004000                          WRL=    4000                    ;WRITE LOCK (BIT #11)
010000                          MOL=    10000                   ;MEDIUM ON-LINE (BIT #12)
020000                          PIP=    20000                   ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                          ERR=    40000                   ;COMPOSIT ERROR. (BIT #14)
100000                          ATA=    100000                  ;ATTENTION ACTIVE (BIT #15)

                                ;ERROR REGISTER #01 (RHER1) (#02)

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

                                ;MAINTAINABILITY REGISTER (RHMR)(#03)
```

```
1302
1303            000001          DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
1304            000002          MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
1305            000004          MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
1306            000010          MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1307            000020          MRD=    20              ;MAINTAINABILITY READ (BIT #4)
1308            000040          MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
1309            000200          DTSY=   200             ;MAINTAINABILITY SYNC DETECTED (BIT #7)
1310
1311                            ;ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1312
1313            000001          AT0=    1               ;DEVICE 0 (BIT #0)
1314            000002          AT1=    2               ;DEVICE 1 (BIT #1)
1315            000004          AT2=    4               ;DEVICE 2 (BIT #2)
1316            000010          AT3=    10              ;DEVICE 3 (BIT #3)
1317            000020          AT4=    20              ;DEVICE 4 (BIT #4)
1318            000040          AT5=    40              ;DEVICE 5 (BIT #5)
1319            000100          AT6=    100             ;DEVICE 6 (BIT #6)
1320            000200          AT7=    200             ;DEVICE 7 (BIT #7)
1321
1322                            ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDA) (#05)
1323                            ;EACH BIT IS CALLED BY BIT NUMBER
1324
1325                            ;DRIVE TYPE REGISTER (RHDT) (#06)
1326                            ;EACH BIT IS CALLED BY BIT NUMBER
1327
1328                            ;LOOK-AHEAD REGISTER (RHLA) (#07)
1329
1330            000001          EXT1=   1               ;EXTENSION 1 (BIT #0)
1331            000002          EXT2=   2               ;EXTENSION 2 (BIT #1)
1332            000004          EXT4=   4               ;EXTENSION 3 (BIT #2)
1333            000010          EXT10=  10              ;EXTENSION 4 (BIT #3)
1334            000020          EXT20=  20              ;EXTENSION 5 (BIT #4)
1335            000040          EXT40=  40              ;EXTENSION 6 (BIT #5)
1336            000100          SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
1337            000200          SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
1338            000400          SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
1339            001000          SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
1340            002000          SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
1341            004000          TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
1342            010000          TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
1343            020000          TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
1344            040000          TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
1345            100000          TRK20=  100000          ;TRACK FIELD 5 (BIT #15)
1346
1347                            ;ERROR REGISTER #2 (RHER2) (#10)
1348
1349            000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1350            000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1351            000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1352            000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1353            000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
1354            000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1355            000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1356            000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
1357            000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
```

```
1358            001000                  MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
1359            002000                  NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
1360            004000                  IXE=    4000                    ;INDEX ERROR (BIT #11)
1361            010000                  VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
1362            020000                  PLU=    20000                   ;PLO UNSAFE (BIT #13)
1363            100000                  ACU=    100000                  ;AC UNSAFE (BIT #15)
1364
1365                                    ;OFFSET REGISTER (RHOF) (#11)
1366
1367            000001                  OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
1368            000002                  OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
1369            000004                  OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
1370            000010                  OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
1371            000020                  OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
1372            000040                  OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
1373            000200                  OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1374            002000                  HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
1375            004000                  ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1376            010000                  FMT22=  10000                   ;FORMAT BIT (BIT #12
1377
1378                                    ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
1379                                    ;EACH BIT IS CALLED BY BIT NUMBER.
1380
1381                                    ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
1382                                    ;EACH BIT IS CALLED BY BIT NUMBER
1383
1384                                    ;SERIAL NUMBER REGISTER (RHSN) (#14)
1385                                    ;EACH IS CALLED BY BIT NUMBER
1386
1387                                    ;ERROR REGISTER #03 (RHER3) (#15)
1388
1389            000001                  PSU=    1                       ;PACK SPEED UNSAFE (BIT #0)
1390            000002                  VUF=    2                       ;VELOCITY UNSAFE (BIT #1)
1391            000010                  UWR=    10                      ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1392            000020                  PRE=    20                      ;DISK PACK ROTATION ERROR (BIT #4)
1393            000040                  ACL=    40                      ;AC LOW (BIT #5)
1394            000100                  DCL=    100                     ;DC LOW (BIT #6)
1395            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1396            100000                  OCYL=   100000                  ;OFF CYLINDER (BIT #15)
1397
1398                                    ;ECC POSITION REGISTER (RHEC1) (#16)
1399                                    ;EACH BIT IS CALLED BY BIT NUMBER
1400
1401                                    ;ECC PATTERN REGISTER (RHEC2) (#17)
1402                                    ;EACH BIT IS CALLED BY BIT NUMBER
1403
1404                                    ;;*****************************************************************************
1405
1406                                    .SBTTL  DEFINITIONS OF THE RH11/RP04 ADDRESS INDEXES
1407
1408                                    ;;*****************************************************************************
1409
1410            000000                  RHCS1=0                         ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
1411            000002                  RHWC=2                          ;WORD COUNT REGISTER (NOT A DRIVE REG)
1412            000004                  RHBA=4                          ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
1413            000006                  RHDA=6                          ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
```

```
1414            000010                  RHCS2=10                ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
1415            000012                  RHDS1=12                ;DRIVE STATUS REGISTER (DRIVE REG 01)
1416            000014                  RHER1=14                ;ERROR REGISTER #1 (DRIVE REG. 02)
1417            000016                  RHAS=16                 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
1418            000020                  RHLA=20                 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
1419            000022                  RHDB=22                 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
1420            000024                  RHMR=24                 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
1421            000026                  RHDT=26                 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
1422            000030                  RHSN=30                 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
1423            000032                  RHOF=32                 ;OFFSET REGISTER (DRIVE REG. 11)
1424            000034                  RHCA=34                 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
1425            000036                  RHCC=36                 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
1426            000040                  RHER2=40                ;ERROR REGISTER #2 (DRIVE REG. 14)
1427            000042                  RHER3=42                ;ERROR REGISTER #3 (DRIVE REG. 15)
1428            000044                  RHEC1=44                ;ECC POSITION REGISTER (DRIVE REG. 16)
1429            000046                  RHEC2=46                ;ECC PATTERN REGISTER (DRIVE REG.  17)
1430
1431
1432                                    .SBTTL  TRAP CATCHER
1433
1434            000000                          .=0
1435                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1436                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1437                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1438
1439                                    .SBTTL  STARTING ADDRESS(ES)
1440            000200                  .=200
1441
1442  000200   000137  002044                  JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
1443                                    ;*STARTING ADDRESS IS LOCATION 200
1444
1445  000204   000137  002410                  JMP     EXEC            ;RESTART
1446                                    ;*RESTART ADDRESS IS LOCATION 204
1447
1448  000210   000137  002560                  JMP     CHANGE          ;CHANGE RH11 ADDRESS AND START
1449                                    ;*START AT LOCATION 210 TO CHANGE THE RH11 ADDRESS FROM 176700
1450
1451
1452
```

```
                                ;*****************************************************************

                                .SBTTL  COMMON TAGS

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

                        000046          .=46
000046  057626                  $ENDAD                  ;;LOGICAL END OF PROGRAM

                        001100          .=1100

001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS  OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS  OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000  000000  000000          .WORD   0,0,0   ;;RESERVED--NOT TO BE USED
001136  177560          $TKS:   177560                  ;;TTY KBD STATUS
001140  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001142  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG.
001144  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG.
001146  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001147  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001151  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152  000000          $REGAD: .WORD   0               ;;CONTAINS THE  FROM
                                                        ;;WHICH  ($REGO) WAS OBTAINED
001154  000000          $REGO:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001156  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001160  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001170  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001172  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR
001174  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001200  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001201  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001202  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
001204  172540          $LKCSR: .WORD   172540          ;ADDR OF KW11-P STATUS REGISTER
001206  172542          $LKCSB: .WORD   172542          ;ADDR OF KW11-P COUNTER BUFFER
001210  000104          $LPVEC: .WORD   104             ;ADDR OF KW11-P VECTOR
001212  177546          $LKS:   .WORD   177546          ;ADDR OF KW11-L STATUS REGISTER
001214  000100          $LLVEC: .WORD   100             ;ADDR OF KW11-L VECTOR
001216  000000          PORTA:  .WORD   0               ;ADDRESS OF PORT A
```

```
1509  001220  000000                PORTB:  .WORD   0               ;ADDRESS OF PORT B
1510  001222  000000                PORTC:  .WORD   0               ;ADDRESS OF DIFFERENT DRIVE
1511  001224  000000                ASR1:   .WORD   0               ;ATA-A OR ATA-B = 1
1512  001226  000000                PTNBR:  .WORD   0               ;CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
1513  001230  000000                SEIZPT: .WORD   0               ;CONTAINS THE ADDRESS OF THE SEIZING PORT
1514  001232  000000                OPPRT:  .WORD   0               ;CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
1515  001234  000000                TSTNUM: .WORD   0               ;NUMBER OF THE CURRENT TEST
1516  001236  000000                CKERR:  .WORD   0               ;IF -1, A REGISTER MISCOMPARISON OCCURED
1517  001240  000000                NOSEIZ: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
1518  001242  000000                RELERR: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
1519  001244  000000                TIME:   .WORD   0               ;ELAPSED TIME COUNTER
1520  001246  000000                WATCH:  .WORD   0               ;WATCH DOG TIMER LOCATION
1521  001250  000000                TIMEA:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
1522  001252  000000                TIMEAP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
1523  001254  000000                TIMEAM: .WORD   0               ;PORT A TIMEOUT VALUE - 25%
1524  001256  000000                TIMEB:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
1525  001260  000000                TIMEBP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
1526  001262  000000                TIMEBM: .WORD   0               ;PORT B TIME VALUE - 25%
1527  001264  000000                TIMES:  .WORD   0               ;STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
1528  001266  000000                KYBCTL: .WORD   0               ;SINGLE TEST INDICATOR
1529
1530                                ;;*****************************************************************************
1531
1532                                .SBTTL  RH11/RP04 UNIBUS AND VECTOR ADDRESSES
1533
1534                                ;;*****************************************************************************
1535
1536  001270  176700                $RPADR: .WORD   176700          ;RH11/RP04 UNIBUS ADDRESS
1537  001272  000254                $RPVEC: .WORD   254             ;RH11 INTERRUPT VECTOR ADDRESS
1538
```

```
                                ;*****************************************************************

                                .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

001274                          $ERRTB:

                                ;ERROR 1

001274  062471                          EM1                     ;WRONG DRIVE TYPE
001276  066464                          DH1
001300  070214                          DT1
001302  070460                          DF1

                                ;ERROR 2

001304  062512                          EM2                     ;DRIVE NOT ON LINE
001306  066464                          DH1
001310  070214                          DT1
001312  070460                          DF1

                                ;ERROR 3

001314  062534                          EM3                     ;SERIAL NUMBERS NOT THE SAME
001316  066535                          DH3
001320  070230                          DT3
001322  070460                          DF1

                                ;ERROR 4

001324  062616                          EM4                     ;DRIVE NOT SEIZED BY PORT 'N'
001326  066604                          DH4
001330  070276                          DT7
001332  070473                          DF7

                                ;ERROR 5

001334  062647                          EM5                     ;WRONG STATUS SEEN BY THE SEIZING PORT
001336  066727                          DH5
001340  070244                          DT5
001342  070465                          DF5

                                ;ERROR 6

001344  062715                          EM6                     ;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA
```

```
1595  001346  067177                          DH13
1596  001350  070316                          DT13
1597  001352  070465                          DF5
1598
1599                                  ;ERROR 7
1600
1601  001354  063015                          EM7             ;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1602  001356  067003                          DH7
1603  001360  070276                          DT7
1604  001362  070473                          DF7
1605
1606                                  ;ERROR 10
1607
1608  001364  063076                          EM10            ;REGISTER CONTENTS INCORRECT
1609  001366  066727                          DH5
1610  001370  070244                          DT5
1611  001372  070465                          DF5
1612
1613                                  ;ERROR 11
1614
1615  001374  063126                          EM11            ;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1616  001376  067126                          DH11
1617  001400  070214                          DT1
1618  001402  070460                          DF1
1619
1620                                  ;ERROR 12
1621
1622  001404  063212                          EM12            ;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1623  001406  067673                          DH36
1624  001410  070404                          DT37
1625  001412  070506                          DF36
1626
1627                                  ;ERROR 13
1628
1629  001414  063262                          EM13            ;'VOLUME VALID' BIT NOT SET BY READIN PRESET
1630  001416  067177                          DH13
1631  001420  070316                          DT13
1632  001422  070465                          DF5
1633
1634                                  ;ERROR 14
1635
1636  001424  063347                          EM14            ;'VOLUME VALID' SET ON THE OPPOSITE PORT
1637  001426  067177                          DH13
1638  001430  070316                          DT13
1639  001432  070465                          DF5
1640
1641                                  ;ERROR 15
1642
1643  001434  063412                          EM15            ;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1644  001436  067003                          DH7
1645  001440  070276                          DT7
1646  001442  070473                          DF7
1647
1648                                  ;ERROR 16
1649
1650  001444  063471                          EM16            ;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
```

```
 1651  001446  067003                          DH7
 1652  001450  070276                          DT7
 1653  001452  070473                          DF7
 1654
 1655                                  ;ERROR 17
 1656
 1657  001454  063544                          EM17            ;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
 1658  001456  067003                          DH7
 1659  001460  070276                          DT7
 1660  001462  070473                          DF7
 1661
 1662                                  ;ERROR 20
 1663
 1664  001464  063623                          EM20            ;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
 1665  001466  067673                          DH36
 1666  001470  070404                          DT37
 1667  001472  070506                          DF36
 1668
 1669                                  ;ERROR 21
 1670
 1671  001474  063703                          EM21            ;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
 1672  001476  067673                          DH36
 1673  001500  070404                          DT37
```

```
1674  001502  070506                          DF36
1675
1676                                  ;ERROR 22
1677
1678  001504  063756                          EM22            ;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
1679  001506  067317                          DH22
1680  001510  070334                          DT22
1681  001512  070502                          DF31
1682
1683                                  ;ERROR 23
1684
1685  001514  064043                          EM23            ;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
1686  001516  067415                          DH23
1687  001520  070346                          DT23
1688  001522  070460                          DF1
1689
1690                                  ;ERROR 24
1691
1692  001524  064111                          EM24            ;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1693  001526  067415                          DH23
1694  001530  070346                          DT23
1695  001532  070460                          DF1
1696
1697
1698                                  ;ERROR 25
1699
1700  001534  064170                          EM25            ;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1701  001536  067673                          DH36
1702  001540  070374                          DT36
1703  001542  070506                          DF36
1704
1705
1706                                  ;ERROR 26
1707
1708  001544  064233                          EM26            ;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
1709  001546  067317                          DH22
1710  001550  070334                          DT22
1711  001552  070502                          DF31
1712
1713                                  ;ERROR 27
1714
1715  001554  064320                          EM27            ;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1716  001556  067003                          DH7
1717  001560  070276                          DT7
1718  001562  070473                          DF7
1719
1720                                  ;ERROR 30
1721
1722  001564  064376                          EM30            ;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1723  001566  067673                          DH36
1724  001570  070374                          DT36
1725  001572  070506                          DF36
1726
1727                                  ;ERROR 31
1728
1729  001574  064473                          EM31            ;DRIVE NOT SEIZED BY PORT AFTER RELEASE WIITH REQUEST SE
```

```
1730  001576  067574                          DH31
1731  001600  070362                          DT31
1732  001602  070502                          DF31
1733
1734                                ;ERROR 32
1735
1736  001604  064550                          EM32            ;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1737  001606  066727                          DH5
1738  001610  070244                          DT5
1739  001612  070465                          DF5
1740
1741                                ;ERROR 33
1742
1743  001614  064621                          EM33            ;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1744  001616  067673                          DH36
1745  001620  070374                          DT36
1746  001622  070506                          DF36
1747
1748                                ;ERROR 34
1749
1750  001624  064723                          EM34            ;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1751  001626  067673                          DH36
1752  001630  070374                          DT36
1753  001632  070506                          DF36
1754
1755                                ;ERROR 35
1756
1757  001634  065026                          EM35            ;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
1758  001636  067673                          DH36
1759  001640  070404                          DT37
1760  001642  070506                          DF36
1761
1762                                ;ERROR 36
1763
1764  001644  065105                          EM36            ;TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS
1765  001646  067673                          DH36
1766  001650  070374                          DT36
1767  001652  070506                          DF36
1768
1769                                ;ERROR 37
1770
1771  001654  065156                          EM37            ;DRIVE IS NON-EXISTANT
1772  001656  067673                          DH36
1773  001660  070404                          DT37
1774  001662  070506                          DF36
1775
1776                                ;ERROR 40
1777
1778  001664  065224                          EM40            ;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1779  001666  066464                          DH1
1780  001670  070346                          DT23
1781  001672  070460                          DF1
1782
1783                                ;ERROR 41
1784
1785  001674  065301                          EM41            ;TIMEOUT CLEARED ATTENTION BIT
```

```
1786  001676  067415                  DH23
1787  001700  070346                  DT23
1788  001702  070460                  DF1
1789
1790                            ;ERROR 42
1791
1792  001704  065343                  EM42            ;DRIVE NOT IN NEUTRAL OR SEIZED
1793  001706  067722                  DH42
1794  001710  070414                  DT42
1795  001712  070511                  DF42
1796
1797                            ;ERROR 43
1798
1799  001714  065431                  EM43            ;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1800  001716  067722                  DH42
1801  001720  070414                  DT42
1802  001722  070511                  DF42
1803
1804                            ;ERROR 44
1805
1806  001724  065506                  EM44            ;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1807  001726  067741                  DH44
1808  001730  070362                  DT31
1809  001732  070502                  DF31
1810
1811                            ;ERROR 45
1812
1813  001734  065563                  EM45            ;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1814  001736  066464                  DH1
1815  001740  070214                  DT1
1816  001742  070460                  DF1
1817
1818                            ;ERROR 46
1819
1820  001744  065642                  EM46            ;CAN'T ACCESS DRIVE THROUGH EITHER PORT
1821  001746  070037                  DH46
1822  001750  070422                  DT46
1823  001752  070502                  DF31
1824
1825                            ;ERROR 47
1826
1827  001754  065711                  EM47            ;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
1828  001756  067415                  DH23
1829  001760  070346                  DT23
1830  001762  070460                  DF1
1831
1832                            ;ERROR 50
1833
1834  001764  065777                  EM50            ;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
1835  001766  067177                  DH13
1836  001770  070316                  DT13
1837  001772  070465                  DF5
1838
1839                            ;ERROR 51
1840
1841  001774  066062                  EM51            ;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
```

```
1842  001776  066727                          DH5
1843  002000  070244                          DT5
1844  002002  070465                          DF5
1845
1846                                  ;ERROR 52
1847
1848  002004  066145                          EM52            ;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1849  002006  067177                          DH13
1850  002010  070316                          DT13
1851  002012  070465                          DF5
1852
1853                                  ;ERROR 53               ;CAN'T READ ATTN BIT FROM OPPOSITE PORT
1854
1855  002014  066243                          EM53
1856  002016  067415                          DH23
1857  002020  070214                          DT1
1858  002022  070460                          DF1
1859
1860                                  ;ERROR 54
1861
1862  002024  066324                          EM54            ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1863  002026  067317                          DH22
1864  002030  070434                          DT54
1865  002032  070502                          DF31
1866
1867                                  ;ERROR 55
1868
1869  002034  066417                          EM55            ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1870  002036  070135                          DH55
1871  002040  070446                          DT55
1872  002042  070513                          DF55
1873
1874
1875
1876                                  ;;**************************************************************************
1877
1878                                  .SBTTL  STARTUP AND INITIALIZATION ROUTINES
1879
1880                                  ;;**************************************************************************
1881
1882  002044                          START:
1883  002044  012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
1884  002052  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1885  002056  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1886  002060  022706  001136                  CMP     #$TKS,R6        ;;DONE?
1887  002064  001374                          BNE     .-6             ;;LOOP BACK IF NO
1888  002066  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1889  002072  012737  057662  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1890  002100  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1891  002106  012737  060040  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1892  002114  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1893  002122  012737  061650  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1894  002130  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1895  002136  013737  057556  057550          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1896  002144  005037  001170                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1897  002150  005037  001172                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
```

```
1898  002154  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1899  002162  012737  002162  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1900  002170  012737  002170  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1901  002176  000005                          RESET                   ;CLEAR THE SYSTEM
1902  002200  104400  061714          START1: TYPE    ,TITLE          ;TYPE PROGRAM NAME
1903  002204  012737  000240  002200          MOV     #NOP,START1     ;DISABLE TITLE TYPEOUT AFTER INITIAL START
1904  002212  012737  000240  002202          MOV     #NOP,START1+2   ;FROM LOCATION 200 OR 210
1905  002220  104400  062012          1$:     TYPE    ,ENTERA         ;ENTER DRIVE ADDRESS
1906  002224  104416                          RDOCT                   ;GET THE ADDRESS
1907  002226  012637  001216                  MOV     (SP)+,PORTA     ;STORE THE ADDRESS
1908  002232  023727  001216  000007          CMP     PORTA,#7        ;SEE IF ADDRESS TOO LARGE
1909  002240  101403                          BLOS    2$              ;BR IF NOT
1910  002242  104400  062042                  TYPE    ,ADRERR         ;TYPE ADDRESS ERROR MESSAGE
1911  002246  000764                          BR      1$              ;TRY AGAIN
1912  002250  013737  001216  001220  2$:     MOV     PORTA,PORTB     ;GENERATE THE PORT B ADDRESS
1913  002256  005237  001220                  INC     PORTB           ;INCREMENT THE ADDRESS
1914  002262  042737  000006  001220          BIC     #6,PORTB        ;LEAVE BIT 0
1915  002270  013746  001216                  MOV     PORTA,-(SP)     ;PUT PORT A ADDRESS ON THE STACK
1916  002274  042716  177771                  BIC     #^C6,(SP)       ;SAVE BITS 1 & 2
1917  002300  052637  001220                  BIS     (SP)+,PORTB     ;SET BITS 1 & 2 IN PORT B ADDRESS
1918  002304  104400  062064                  TYPE    ,PORTAIS        ;'PORT A ADDRESS IS '
1919  002310  013746  001216                  MOV     PORTA,-(SP)     ;PUT THE ADDRESS ON THE STACK
1920  002314  104410                          TYPDS                   ;TYPE PORT A ADDRESS
1921  002316  104400  062112                  TYPE    ,PORTBIS        ;'PORT B ADDRESS IS ,
1922  002322  013746  001220                  MOV     PORTB,-(SP)     ;PUT ADDRESS ON THE STACK
1923  002326  104410                          TYPDS                   ;TYPE PORT B ADDRESS
1924  002330  104400  001201                  TYPE    ,$CRLF          ;ANOTHER CR-LF
1925  002334  013737  001216  001222          MOV     PORTA,PORTC     ;GENERATE ADDRESS OF DRIVE NOT TESTED
1926  002342  062737  000006  001222          ADD     #6,PORTC        ;COMPLEMENT SOME BITS
1927  002350  042737  177770  001222          BIC     #^C7,PORTC      ;SAVE ONLY LOWER BITS
1928  002356  013701  001216                  MOV     PORTA,R1        ;USE PORT A ADDRESS AS INDEX
1929  002362  116137  070636  001224          MOVB    ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
1930  002370  004737  057264                  JSR     PC,CKCLK        ;SETUP CLOCK
1931  002374  000137  002410                  JMP     EXEC            ;CLOCK HAS BEEN STARTED
1932  002400  104400  062140                  TYPE    ,NOCLOCK        ;NO CLOCK ON SYSTEM
1933  002404  000000                          HALT                    ;FATAL ERROR
1934  002406  000776                          BR      .-2             ;INTERLOCK HALT
1935
1936                                  ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
1937
1938  002410  000005                  EXEC:   RESET                   ;CLEAR EVERYTHING
1939  002412  012737  000240  177776          MOV     #<5*32.>,PS     ;SET PROCESSOR PRIORITY TO 5
1940  002420  104400  001201                  TYPE    ,$CRLF          ;CR-LF
1941  002424  013700  001270                  MOV     $RPADR,R0       ;RH11 ADDRESS FOR INDEXING
1942  002430  012706  001100                  MOV     #STACK,R6       ;LOAD STACK POINTER
1943  002434  004737  057264                  JSR     PC,CKCLK        ;START THE CLOCK
1944  002440  000240                          NOP                     ;RETURN IF NO CLOCK
1945  002442  005037  001266                  CLR     KYBCTL          ;CLEAR SINGLE TEST INDICATOR
1946  002446  005037  001100                  CLR     $PASS           ;CLEAR THE PASS COUNT
1947  002452  112737  000001  001115          MOVB    #1,$ERMAX       ;SET ERROR MAX TO 1
1948  002460  012737  002460  001106          MOV     #.,$LPADR       ;INITIAL SETTING FOR LOOP ADDRESS
1949  002466  012737  002466  001110          MOV     #.,$LPERR       ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
1950  002474  104400  062207                  TYPE    ,TESTNO         ;ASK FOR TEST NUMBER
1951  002500  104416                          RDOCT                   ;GET THE NUMBER
1952  002502  012601                          MOV     (SP)+,R1        ;PUT ENTRY INTO R1
1953  002504  001002                          BNE     .+6             ;BR IF NOT ZERO
```

```
1954   002506  000137  002710                  JMP     TST1            ;ENTER ZERO - PERFORM ALL TESTS
1955   002512  020137  070646                  CMP     R1,MAXTN        ;SEE IF NUMBER GREATER THAN MAXIMUM
1956   002516  003403                          BLE     1$              ;BR IF LESS OR EQUAL
1957   002520  104400  062227                  TYPE    ,BADNO          ;BAD ENTRY
1958   002524  000731                          BR      EXEC            ;TRY AGAIN
1959   002526  005301                  1$:     DEC     R1              ;DECREMENT ENTRY
1960   002530  006301                          ASL     R1              ;SHIFT IT LEFT
1961   002532  016137  070520  002556          MOV     TSTADR(R1),2$   ;GET THE TEST ADDRESS
1962   002540  005237  001266                  INC     KYBCTL          ;SET SINGLE TEST INDICATOR
1963   002544  012737  000001  001104          MOV     #1,$ICNT        ;PRESET ITERATION COUNT
1964   002552  000177  000000                  JMP     @2$             ;GO TO THE SELECTED TEST
1965   002556  000000                  2$:     .WORD   0               ;TEST ADDRESS GOES HERE
1966
1967                                  ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
1968
1969   002560  000005                  CHANGE: RESET                   ;CLEAR THE SYSTEM
1970   002562  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT ALL INTERRUPTS
1971   002570  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
1972   002574  012737  061650  000034          MOV     #$TRAP,@#TRAPVEC  ;LOAD TRAP VECTOR
1973   002602  012737  000340  000036          MOV     #340,@#TRAPVEC+2  ;LEVEL 7
1974   002610  104400  062267                  TYPE    ,ADDRIS         ;TYPE OUT WHAT THE PRESENT ADDRESS IS
1975   002614  013746  001270                  MOV     $RPADR,-(SP)    ;PUT THE ADDRESS ON THE STACK
1976   002620  104402                          TYPOC                   ;TYPE THE ACTUAL ADDRESS
1977   002622  104400  001201                  TYPE    ,$CRLF          ;CR-LF
1978   002626  104400  062347                  TYPE    ,NTRH11         ;ASK FOR NEW ADDRESS
1979   002632  104416                          RDOCT
1980   002634  005716                          TST     (SP)            ;0 OR 'CR' ENTERED ?
1981   002636  001402                          BEQ     1$              ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
1982   002640  011637  001270                  MOV     (SP),$RPADR     ;NEW RH11 ADDRESS
1983   002644  012737  002666  000004  1$:     MOV     #2$,4           ;LOAD TRAP ADDRESS
1984   002652  013700  001270                  MOV     $RPADR,R0       ;RH11 ADDRESS
1985   002656  062700  000002                  ADD     #2,R0           ;FORM ADDRESS OF RHWC
1986   002662  005710                          TST     (R0)            ;SEE IF RH11 RESPONDS AT  THAT ADDRESS
1987   002664  000405                          BR      3$              ;BR, RH11 ALIVE AT PRESENT ADDRESS
1988   002666  104400  062401          2$:     TYPE    ,NORESP         ;REPORT NO RESPONSE
1989   002672  010046                          MOV     R0,-(SP)        ;SETUP TO CONVERT THE ADDRESS
1990   002674  104402                          TYPOC                   ;TYPE THE ADDRESS
1991   002676  000730                          BR      CHANGE          ;GET ADDRESS AGAIN
1992   002700  000137  002044          3$:     JMP     START           ;GO TO THE STARTING ADDRESS
1993
1994
1995                                  ;;**************************************************************************
1996
1997                                  .SBTTL  *** TESTS ***
1998
1999                                  ;;**************************************************************************
2000
2001
2002   002704  013700  001270          TST1AA: MOV     $RPADR,R0       ;;RESTORE R0 AFTER END OF PASS
2003
2004                                  ;*********************************************************************
2005                                  ;*TEST 1         DRIVE ACCESS TEST
2006                                  ;*
2007                                  ;*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
2008                                  ;*
2009                                  ;*  A.  SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
```

```
                                        ;*      DRIVE IS A DUAL PORT RP04, 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.
                                        ;*
                                        ;*****************************************************************
002710                                  TST1:
002710  000004                                SCOPE                   ;INITIALIZE THE SCOPE HANDLER
002712  005737  001266                      TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
002716  001406                              BEQ     2$              ;BR IF NOT
002720  100002                              BPL     1$              ;BR IF JUST ENTERED TEST
002722  000137  002410                      JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
002726  012737  177777  001266  1$:        MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
002734  112737  000001  001102  2$:        MOVB    #1,$TSTNM       ;TEST NUMBER
002742  012737  002764  001106              MOV     #TEST1,$LPADR   ;LOAD LOOP ON TEST ADDRESS
002750  012737  002764  001110              MOV     #TEST1,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
002756  012737  000001  001170              MOV     #1,$TIMES       ;;DO 1 ITERATION

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

002764                                  TEST1:
                                        ;;*************************************************************************
                                        ;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

002764  113760  001216  000010                  MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
002772  013737  001216  001226                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003000  005760  000012                          TST     RHDS1(R0)        ;SEE IF DRIVE (PORT A) PRESENT
003004  005037  001236                          CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
003010  016037  000010  001126                  MOV     RHCS2(R0),$BDDAT  ;GET CONTENTS OF RHCS2
003016  012737  000010  001122                  MOV     #RHCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003024  060037  001122                          ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
003030  005037  001124                          CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
003034  013737  001126  001156                  MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
003042  042737  167777  001156                  BIC     #^CNED,$TMP0  ;SAVE SPECIFIED BITS
003050  023737  001124  001156                  CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
003056  001414                                  BEQ     64$              ;BR IF OK
003060  013737  001126  001166                  MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
003066  042737  010000  001166                  BIC     #NED,$TMP4       ;CLEAR THE MASKED BITS
003074  053737  001166  001124                  BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
003102  104037                                  ERROR   37               ;TYPE MESSAGE 37
003104  005137  001236                          COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
003110  000240                          64$:    NOP
003112  005737  001236                          TST     CKERR            ;WAS 'NED' SET ?
003116  001403                                  BEQ     .+10             ;BR IF NOT
003120  012760  000040  000010                  MOV     #CLR,RHCS2(R0)   ;ISSUE MASSBUS INIT TO CLEAR 'NED'
003126  113760  001220  000010                  MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
003134  013737  001220  001226                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003142  005760  000012                          TST     RHDS1(R0)        ;SEE IF DRIVE (PORT B) PRESENT
003146  005037  001236                          CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
003152  016037  000010  001126                  MOV     RHCS2(R0),$BDDAT  ;GET CONTENTS OF RHCS2
003160  012737  000010  001122                  MOV     #RHCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003166  060037  001122                          ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
003172  005037  001124                          CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
```

```
2066  003176  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2067  003204  042737  167777  001156          BIC     #↑CNED,$TMP0  ;SAVE SPECIFIED BITS
2068  003212  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2069  003220  001414                          BEQ     65$             ;BR IF OK
2070  003222  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2071  003230  042737  010000  001166          BIC     #NED,$TMP4      ;CLEAR THE MASKED BITS
2072  003236  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2073  003244  104037                          ERROR   37              ;TYPE MESSAGE 37
2074  003246  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2075  003252  000240                  65$:    NOP
2076  003254  005737  001236                  TST     CKERR           ;WAS 'NED' SET ?
2077  003260  001403                          BEQ     .+10            ;BR IF NOT
2078  003262  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2079
2080                                  ;;*************************************************************************
2081                                  ;CONFIRM THAT DRIVE IS AN RP04 AND IS DUAL PORT
2082
2083  003270  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2084  003276  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2085  003304  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2086  003310  016037  000026  001126          MOV     RHDT(R0),$BDDAT  ;GET CONTENTS OF RHDT
2087  003316  012737  000026  001122          MOV     #RHDT,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2088  003324  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2089  003330  012737  024020  001124          MOV     #24020,$GDDAT  ;WHAT REGISTER SHOULD BE
2090  003336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
2091  003344  001403                          BEQ     66$             ;BR IF OK
2092  003346  104001                          ERROR   1               ;TYPE MESSAGE 1
2093  003350  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2094  003354  000240                  66$:    NOP
2095  003356  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2096  003364  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2097  003372  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2098  003376  016037  000026  001126          MOV     RHDT(R0),$BDDAT  ;GET CONTENTS OF RHDT
2099  003404  012737  000026  001122          MOV     #RHDT,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2100  003412  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2101  003416  012737  024020  001124          MOV     #24020,$GDDAT  ;WHAT REGISTER SHOULD BE
2102  003424  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
2103  003432  001403                          BEQ     67$             ;BR IF OK
2104  003434  104001                          ERROR   1               ;TYPE MESSAGE 1
2105  003436  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2106  003442  000240                  67$:    NOP
2107
2108                                  ;;*************************************************************************
2109                                  ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
2110
2111  003444  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2112  003452  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2113  003460  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2114  003464  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2115  003472  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2116  003500  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2117  003504  012737  001000  001124          MOV     #PGM,$GDDAT  ;WHAT REGISTER SHOULD BE
2118  003512  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2119  003520  042737  176777  001156          BIC     #↑CPGM,$TMP0  ;SAVE SPECIFIED BITS
2120  003526  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2121  003534  001414                          BEQ     68$             ;BR IF OK
```

```
2122  003536  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2123  003544  042737  001000  001166          BIC     #PGM,$TMP4      ;CLEAR THE MASKED BITS
2124  003552  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2125  003560  104045                          ERROR   45              ;TYPE MESSAGE 45
2126  003562  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2127  003566  000240                  68$:    NOP
2128  003570  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2129  003574  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2130  003602  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2131  003610  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2132  003614  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2133  003622  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2134  003630  042737  167177  001156          BIC     #^C10600,$TMP0  ;SAVE SPECIFIED BITS
2135  003636  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2136  003644  001414                          BEQ     69$             ;BR IF OK
2137  003646  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2138  003654  042737  010600  001166          BIC     #10600,$TMP4    ;CLEAR THE MASKED BITS
2139  003662  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2140  003670  104002                          ERROR   2               ;TYPE MESSAGE 2
2141  003672  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2142  003676  000240                  69$:    NOP
2143  003700  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2144  003706  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2145  003714  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2146  003720  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2147  003726  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2148  003734  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2149  003740  012737  001000  001124          MOV     #PGM,$GDDAT     ;WHAT REGISTER SHOULD BE
2150  003746  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2151  003754  042737  176777  001156          BIC     #^CPGM,$TMP0    ;SAVE SPECIFIED BITS
2152  003762  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2153  003770  001414                          BEQ     70$             ;BR IF OK
2154  003772  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2155  004000  042737  001000  001166          BIC     #PGM,$TMP4      ;CLEAR THE MASKED BITS
2156  004006  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2157  004014  104045                          ERROR   45              ;TYPE MESSAGE 45
2158  004016  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2159  004022  000240                  70$:    NOP
2160  004024  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2161  004030  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2162  004036  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2163  004044  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2164  004050  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2165  004056  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2166  004064  042737  167177  001156          BIC     #^C10600,$TMP0  ;SAVE SPECIFIED BITS
2167  004072  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2168  004100  001414                          BEQ     71$             ;BR IF OK
2169  004102  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2170  004110  042737  010600  001166          BIC     #10600,$TMP4    ;CLEAR THE MASKED BITS
2171  004116  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2172  004124  104002                          ERROR   2               ;TYPE MESSAGE 2
2173  004126  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2174  004132  000240                  71$:    NOP
2175
2176                                          ;;*************************************************************************
2177                                          ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
```

```
2178
2179  004134  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2180  004142  016037  000030  001124          MOV     RHSN(R0),$GDDAT  ;STORE THE PORT A SERIAL NUMBER
2181  004150  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2182  004156  016037  000030  001126          MOV     RHSN(R0),$BDDAT  ;STORE THE PORT B SERIAL NUMBER
2183  004164  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ARE THEY THE SAME ?
2184  004172  001406                          BEQ     1$               ;BR IF THEY ARE
2185  004174  104003                          ERROR   3                ;REPORT THE ERROR
2186  004176  032737  100000  177570          BIT     #SW15,SWR        ;HALT ON ERROR ?
2187  004204  001001                          BNE     1$               ;BR IF SET - PROGRAM HAS ALREADY HALTED
2188  004206  000000                          HALT                     ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
2189  004210                          1$:
2190
2191                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2192
2193  004210  105737  001103                  TSTB    $ERFLG           ;DID AN ERROR OCCUR ?
2194  004214  001412                          BEQ     TST2             ;;BR IF NOT
2195  004216  032737  001000  177570          BIT     #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
2196  004224  001406                          BEQ     TST2             ;;BR IF NOT
2197  004226  105037  001103                  CLRB    $ERFLG           ;CLEAR THE ERROR FLAG
2198  004232  005037  001170                  CLR     $TIMES           ;CLEAR THE MAX ITERATION COUNT
2199  004236  000177  174646                  JMP     @$LPERR          ;GO TO THE LOOP ADDRESS
2200
2201
2202                                  ;*****************************************************************
2203                                  ;*TEST 2        PORT 'A' SEIZE/TIMEOUT TEST
2204                                  ;*
2205                                  ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
2206                                  ;*      THE DRIVE.
2207                                  ;*
2208                                  ;*  A.  WRITE 0'S INTO RHDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE
2209                                  ;*      HAS BEEN SEIZED.
2210                                  ;*
2211                                  ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RHCS1, THROUGH PORT 'B';
2212                                  ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
2213                                  ;*
2214                                  ;*  C.  WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
2215                                  ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
2216                                  ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
2217                                  ;*      NEUTRAL.
2218                                  ;*
2219                                  ;*****************************************************************
2220  004242                          TST2:
2221  004242  000004                          SCOPE                    ;INITIALIZE THE SCOPE HANDLER
2222  004244  005737  001266                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TESTS ?
2223  004250  001406                          BEQ     2$               ;BR IF NOT
2224  004252  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
2225  004254  000137  002410                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
2226  004260  012737  177777  001266  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
2227  004266  112737  000002  001102  2$:     MOVB    #2,$TSTNM        ;TEST NUMBER
2228  004274  012737  004316  001106          MOV     #TEST2,$LPADR    ;LOAD LOOP ON TEST ADDRESS
2229  004302  012737  004316  001110          MOV     #TEST2,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
2230  004310  012737  000012  001170          MOV     #10.,$TIMES      ;;DO 10. ITERATIONS
2231
2232
2233                                  ;;*****************************************************************
```

```
2234                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
2235
2236  004316                            TEST2:
2237  004316  012737  000240  177776            MOV     #(5*32.),@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
2238  004324  005037  001250                    CLR     TIMEA           ;CLEAR TIMEOUT VALUE FOR PORT A
2239  004330  005037  001252                    CLR     TIMEAP  ;CLEAR UPPER TIMEOUT TOLERANCE
2240  004334  005037  001254                    CLR     TIMEAM  ;CLEAR LOWER TIMEOUT TOLERANCE
2241
2242                                    ;;*************************************************************************
2243                                    ;START THE TIMER
2244
2245  004340  005037  001244                    CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
2246  004344  012737  003720  001246            MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
2247
2248                                    ;;*************************************************************************
2249
2250                                            ;SEIZE THE DRIVE THROUGH PORT A
2251
2252  004352  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
2253  004360  013737  001216  001230            MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
2254  004366  005060  000012                    CLR     RHDS1(R0)       ;WRITE RHDS1
2255  004372  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
2256  004400  013737  001220  001226            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2257  004406  013737  001220  001232            MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
2258  004414  016037  000012  001126            MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
2259  004422  010037  001122                    MOV     R0,$BDADR       ;RH11 BASE ADDRESS
2260  004426  062737  000012  001122            ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
2261  004434  005037  001124                    CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
2262  004440  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
2263  004446  001403                            BEQ     .+10            ;BR IF IT IS
2264  004450  104004                            ERROR   4               ;REPORT THE ERROR
2265  004452  000137  005634                    JMP     5$              ;BYPASS REST OF THE SUBTEST
2266  004456  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
2267  004464  013737  001216  001226            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2268  004472  016037  000012  001126            MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2269  004500  012737  011600  001124            MOV     #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2270  004506  013737  001124  001160            MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
2271  004514  005137  001160                    COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
2272  004520  013737  001126  001156            MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
2273  004526  043737  001160  001156            BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
2274  004534  023737  001124  001156            CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
2275  004542  001401                            BEQ     .+4             ;BR IF THEY ARE
2276  004544  104005                            ERROR 5                 ;REPORT THE ERROR
2277
2278                                    ;;*************************************************************************
2279                                    ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2280
2281  004546  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
2282  004554  013737  001220  001226            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2283  004562  016046  000046                    MOV     RHEC2(R0),-(SP) ;STORE REGISTER RHEC2, PORT B, FOR CHECK
2284  004566  016046  000044                    MOV     RHEC1(R0),-(SP) ;STORE REGISTER RHEC1, PORT B, FOR CHECK
2285  004572  016046  000042                    MOV     RHER3(R0),-(SP) ;STORE REGISTER RHER3, PORT B, FOR CHECK
2286  004576  016046  000030                    MOV     RHSN(R0),-(SP)  ;STORE REGISTER RHSN, PORT B, FOR CHECK
2287  004602  016046  000036                    MOV     RHCC(R0),-(SP)  ;STORE REGISTER RHCC, PORT B, FOR CHECK
2288  004606  016046  000034                    MOV     RHCA(R0),-(SP)  ;STORE REGISTER RHCA, PORT B, FOR CHECK
2289  004612  016046  000032                    MOV     RHOF(R0),-(SP)  ;STORE REGISTER RHOF, PORT B, FOR CHECK
```

```
2290  004616  016046  000040                  MOV     RHER2(R0),-(SP) ;STORE REGISTER RHER2, PORT B, FOR CHECK
2291  004622  016046  000020                  MOV     RHLA(R0),-(SP)  ;STORE REGISTER RHLA, PORT B, FOR CHECK
2292  004626  016046  000026                  MOV     RHDT(R0),-(SP)  ;STORE REGISTER RHDT, PORT B, FOR CHECK
2293  004632  016046  000006                  MOV     RHDA(R0),-(SP)  ;STORE REGISTER RHDA, PORT B, FOR CHECK
2294  004636  016046  000024                  MOV     RHMR(R0),-(SP)  ;STORE REGISTER RHMR, PORT B, FOR CHECK
2295  004642  016046  000014                  MOV     RHER1(R0),-(SP) ;STORE REGISTER RHER1, PORT B, FOR CHECK
2296
2297                                  ;;*************************************************************************
2298                                  ;WAIT FOR PORT A TO TIMEOUT
2299
2300  004646  005760  000012          1$:     TST     RHDS1(R0)       ;WAIT FOR THE DRIVE TO TIMEOUT
2301  004652  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
2302  004654  005737  001246                  TST     WATCH           ;CHECK WATCH
2303  004660  001372                          BNE     1$              ;BR IF NOT ZERO
2304  004662  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
2305  004664  000137  005254                  JMP     4$              ;BYPASS TIMEOUT TIME CHECK
2306  004670  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS   ;SET PRIORITY TO 7 TO STOP CLOCK
2307  004676  013737  001244  001250          MOV     TIME,TIMEA      ;SAVE THE ELAPSED TIME FOR PORT A
2308  004704  004537  057450                  JSR     R5,TOLER        ;CALCULATE THE TOLERANCE
2309  004710  001250                          .WORD   TIMEA           ;TIMEOUT VALUE FOR PORT A
2310  004712  012637  001252                  MOV     (SP)+,TIMEAP    ;+25% TOLERANCE
2311  004716  012637  001254                  MOV     (SP)+,TIMEAM    ;-25% TOLERANCE
2312
2313                                  ;;*************************************************************************
2314                                  ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2315
2316  004722  023727  001244  000764          CMP     TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
2317  004730  103001                          BHIS    3$              ;BR IF IT  WAS
2318  004732  104055                          ERROR   55              ;REPORT TIMEOUT TOO SHORT
2319
2320                                  ;;*************************************************************************
2321                                  ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2322
2323  004734  012737  000240  177776  3$:     MOV     #<5*32.>,@#PS   ;RESTORE PRIORITY TO 5
2324
2325                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2326
2327  004742  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
2328  004746  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
2329  004754  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
2330  004760  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
2331  004766  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
2332  004774  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2333  005002  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
2334  005010  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2335  005016  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
2336  005024  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2337  005032  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
2338  005040  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2339  005046  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2340  005054  001006                          BNE     64$             ;BR IF NOT
2341  005056  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2342  005062  001037                          BNE     66$             ;BR IF NOT
2343  005064  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2344  005066  000137  005252                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
2345  005072  013737  001162  001126  64$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
```

```
2346  005100  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2347  005106  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
2348  005114  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2349  005120  001414                          BEQ     65$             ;BR IF ZERO
2350  005122  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2351  005130  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2352  005136  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
2353  005144  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2354  005150  001004                          BNE     66$             ;BR IF NOT
2355  005152  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2356  005160  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
2357  005162  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
2358  005170  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
2359  005176  042737  100100  001162          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2360  005204  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
2361  005212  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
2362  005214  104007                          ERROR   7               ;REPORT ERROR
2363  005216  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
2364  005224  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
2365  005232  042737  100100  001164          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2366  005240  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2367  005246  001401                          BEQ     68$             ;BR IF OK
2368  005250  104007                          ERROR   7               ;REPORT ERROR
2369  005252  000240                  68$:    NOP
2370
2371                                  ;;*****************************************************************
2372                                  ;CHECK THE REGISTERS STORED THROUGH PORT B.  ALL REGISTERS SHOULD BE ZERO.
2373                                  ;THE REGISTERS ARE STORED ON THE STACK.
2374
2375  005254  013737  001220  001226  4$:     MOV     PORTB,PTNBR     ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2376  005262  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER1
2377  005266  062737  000014  001122          ADD     #RHER1,$BDADR   ;ADDRESS OF RHER1 FOR TYPEOUT
2378  005274  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER1
2379  005300  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2380  005302  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2381  005304  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHMR
2382  005310  062737  000024  001122          ADD     #RHMR,$BDADR    ;ADDRESS OF RHMR FOR TYPEOUT
2383  005316  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHMR
2384  005322  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2385  005324  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2386  005326  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHDA
2387  005332  062737  000006  001122          ADD     #RHDA,$BDADR    ;ADDRESS OF RHDA FOR TYPEOUT
2388  005340  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHDA
2389  005344  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2390  005346  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2391  005350  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHDT
2392  005354  062737  000026  001122          ADD     #RHDT,$BDADR    ;ADDRESS OF RHDT FOR TYPEOUT
2393  005362  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHDT
2394  005366  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2395  005370  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2396  005372  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHLA
2397  005376  062737  000020  001122          ADD     #RHLA,$BDADR    ;ADDRESS OF RHLA FOR TYPEOUT
2398  005404  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHLA
2399  005410  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2400  005412  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2401  005414  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER2
```

```
2402  005420  062737  000040  001122          ADD     #RHER2,$BDADR   ;ADDRESS OF RHER2 FOR TYPEOUT
2403  005426  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER2
2404  005432  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2405  005434  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2406  005436  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHOF
2407  005442  062737  000032  001122          ADD     #RHOF,$BDADR    ;ADDRESS OF RHOF FOR TYPEOUT
2408  005450  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHOF
2409  005454  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2410  005456  104006                          ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2411  005460  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHCA
2412  005464  062737  000034  001122          ADD     #RHCA,$BDADR    ;ADDRESS OF RHCA FOR TYPEOUT
2413  005472  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHCA
2414  005476  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2415  005500  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2416  005502  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHCC
2417  005506  062737  000036  001122          ADD     #RHCC,$BDADR    ;ADDRESS OF RHCC FOR TYPEOUT
2418  005514  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHCC
2419  005520  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2420  005522  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2421  005524  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHSN
2422  005530  062737  000030  001122          ADD     #RHSN,$BDADR    ;ADDRESS OF RHSN FOR TYPEOUT
2423  005536  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHSN
2424  005542  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2425  005544  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2426  005546  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER3
2427  005552  062737  000042  001122          ADD     #RHER3,$BDADR   ;ADDRESS OF RHER3 FOR TYPEOUT
2428  005560  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER3
2429  005564  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2430  005566  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2431  005570  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHEC1
2432  005574  062737  000044  001122          ADD     #RHEC1,$BDADR   ;ADDRESS OF RHEC1 FOR TYPEOUT
2433  005602  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHEC1
2434  005606  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2435  005610  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2436  005612  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHEC2
2437  005616  062737  000046  001122          ADD     #RHEC2,$BDADR   ;ADDRESS OF RHEC2 FOR TYPEOUT
2438  005624  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHEC2
2439  005630  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2440  005632  104006                          ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2441  005634                          5$:
2442
2443                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2444
2445  005634  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
2446  005640  001412                          BEQ     TST3            ;;BR IF NOT
2447  005642  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
2448  005650  001406                          BEQ     TST3            ;;BR IF NOT
2449  005652  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
2450  005656  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
2451  005662  000177  173222                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
2452
2453                                  ;*********************************************************
2454                                  ;*TEST 3        PORT 'B' SEIZE/TIMEOUT TEST
2455                                  ;*
2456                                  ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
2457                                  ;*      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.
                                        ;*
                                        ;*****************************************************************
005666                                  TST3:
005666  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
005670  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
005674  001406                                  BEQ     2$              ;BR IF NOT
005676  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
005700  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
005704  012737  177777  001266    1$:           MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
005712  112737  000003  001102    2$:           MOVB    #3,$TSTNM       ;TEST NUMBER
005720  012737  005742  001106                  MOV     #TEST3,$LPADR   ;LOAD LOOP ON TEST ADDRESS
005726  012737  005742  001110                  MOV     #TEST3,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
005734  012737  000012  001170                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

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

005742                                  TEST3:
005742  012737  000240  177776                  MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
005750  005037  001256                          CLR     TIMEB           ;CLEAR TIMEOUT VALUE FOR PORT B
005754  005037  001260                          CLR     TIMEBP  ;CLEAR UPPER TIMEOUT TOLERANCE
005760  005037  001262                          CLR     TIMEBM  ;CLEAR LOWER TIMEOUT TOLERANCE

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

005764  005037  001244                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
005770  012737  C03720  001246                  MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS

                                        ;;*****************************************************************

                                        ;SEIZE THE DRIVE THROUGH PORT B

005776  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
006004  013737  001220  001230                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
006012  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
006016  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
006024  013737  001216  001226                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006032  013737  001216  001232                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
006040  016037  000012  001126                  MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
006046  010037  001122                          MOV     R0,$BDADR       ;RH11 BASE ADDRESS
006052  062737  000012  001122                  ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
006060  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
006064  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
006072  001403                                  BEQ     .+10            ;BR IF IT IS
```

```
2514  006074  104004                          ERROR   4               ;REPORT THE ERROR
2515  006076  000137  007260                  JMP     5$              ;BYPASS REST OF THE SUBTEST
2516  006102  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2517  006110  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2518  006116  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
2519  006124  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;EXPECTED STATUS
2520  006132  013737  001124  001160          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
2521  006140  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
2522  006144  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
2523  006152  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
2524  006160  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
2525  006166  001401                          BEQ     .+4             ;BR IF THEY ARE
2526  006170  104005                          ERROR 5                 ;REPORT THE ERROR
2527
2528                                  ;;**********************************************************************
2529                                  ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2530
2531  006172  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2532  006200  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2533  006206  016046  000046                  MOV     RHEC2(R0),-(SP) ;STORE REGISTER RHEC2, PORT A, FOR CHECK
2534  006212  016046  000044                  MOV     RHEC1(R0),-(SP) ;STORE REGISTER RHEC1, PORT A, FOR CHECK
2535  006216  016046  000042                  MOV     RHER3(R0),-(SP) ;STORE REGISTER RHER3, PORT A, FOR CHECK
2536  006222  016046  000030                  MOV     RHSN(R0),-(SP)  ;STORE REGISTER RHSN, PORT A, FOR CHECK
2537  006226  016046  000036                  MOV     RHCC(R0),-(SP)  ;STORE REGISTER RHCC, PORT A, FOR CHECK
2538  006232  016046  000034                  MOV     RHCA(R0),-(SP)  ;STORE REGISTER RHCA, PORT A, FOR CHECK
2539  006236  016046  000032                  MOV     RHOF(R0),-(SP)  ;STORE REGISTER RHOF, PORT A, FOR CHECK
2540  006242  016046  000040                  MOV     RHER2(R0),-(SP) ;STORE REGISTER RHER2, PORT A, FOR CHECK
2541  006246  016046  000020                  MOV     RHLA(R0),-(SP)  ;STORE REGISTER RHLA, PORT A, FOR CHECK
2542  006252  016046  000026                  MOV     RHDT(R0),-(SP)  ;STORE REGISTER RHDT, PORT A, FOR CHECK
2543  006256  016046  000006                  MOV     RHDA(R0),-(SP)  ;STORE REGISTER RHDA, PORT A, FOR CHECK
2544  006262  016046  000024                  MOV     RHMR(R0),-(SP)  ;STORE REGISTER RHMR, PORT A, FOR CHECK
2545  006266  016046  000014                  MOV     RHER1(R0),-(SP) ;STORE REGISTER RHER1, PORT A, FOR CHECK
2546
2547                                  ;;**********************************************************************
2548                                  ;WAIT FOR PORT B TO TIMEOUT
2549
2550  006272  005760  000012          1$:     TST     RHDS1(R0)       ;WAIT FOR THE DRIVE TO TIMEOUT
2551  006276  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
2552  006300  005737  001246                  TST     WATCH           ;CHECK WATCH
2553  006304  001372                          BNE     1$              ;BR IF NOT ZERO
2554  006306  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
2555  006310  000137  006700                  JMP     4$              ;BYPASS TIMEOUT TIME CHECK
2556  006314  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS   ;SET PRIORITY TO 7 TO STOP CLOCK
2557  006322  013737  001244  001256          MOV     TIME,TIMEB      ;SAVE THE ELAPSED TIME FOR PORT B
2558  006330  004537  057450                  JSR     R5,TOLER        ;CALCULATE THE TOLERANCE
2559  006334  001256                          .WORD   TIMEB           ;TIMEOUT VALUE FOR PORT B
2560  006336  012637  001260                  MOV     (SP)+,TIMEBP    ;+25% TOLERANCE
2561  006342  012637  001262                  MOV     (SP)+,TIMEBM    ;-25% TOLERANCE
2562
2563                                  ;;**********************************************************************
2564                                  ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2565
2566  006346  023727  001244  000764          CMP     TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
2567  006354  103001                          BHIS    3$              ;BR IF IT  WAS
2568  006356  104055                          ERROR   55              ;REPORT TIMEOUT TOO SHORT
2569
```

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

006360  012737  000240  177776  3$:     MOV     #<5*32.>,@#PS   ;RESTORE PRIORITY TO 5

                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL

006366  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
006372  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
006400  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
006404  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
006412  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
006420  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
006426  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
006434  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006442  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
006450  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
006456  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
006464  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006472  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
006500  001006                          BNE     64$             ;BR IF NOT
006502  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
006506  001037                          BNE     66$             ;BR IF NOT
006510  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
006512  000137  006676                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
006516  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
006524  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006532  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
006540  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
006544  001414                          BEQ     65$             ;BR IF ZERO
006546  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006554  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
006562  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
006570  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
006574  001004                          BNE     66$             ;BR IF NOT
006576  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
006604  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
006606  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
006614  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
006622  042737  100100  001162          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
006630  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
006636  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
006640  104007                          ERROR   7               ;REPORT ERROR
006642  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
006650  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
006656  042737  100100  001164          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
006664  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
006672  001401                          BEQ     68$             ;BR IF OK
006674  104007                          ERROR   7               ;REPORT ERROR
006676  000240                  68$:    NOP

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

006700  013737  001216  001226  4$:     MOV     PORTA,PTNBR     ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
```

```
2626  006706  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER1
2627  006712  062737  000014  001122        ADD     #RHER1,$BDADR   ;ADDRESS OF RHER1 FOR TYPEOUT
2628  006720  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER1
2629  006724  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2630  006726  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2631  006730  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHMR
2632  006734  062737  000024  001122        ADD     #RHMR,$BDADR    ;ADDRESS OF RHMR FOR TYPEOUT
2633  006742  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHMR
2634  006746  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2635  006750  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2636  006752  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHDA
2637  006756  062737  000006  001122        ADD     #RHDA,$BDADR    ;ADDRESS OF RHDA FOR TYPEOUT
2638  006764  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHDA
2639  006770  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2640  006772  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2641  006774  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHDT
2642  007000  062737  000026  001122        ADD     #RHDT,$BDADR    ;ADDRESS OF RHDT FOR TYPEOUT
2643  007006  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHDT
2644  007012  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2645  007014  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2646  007016  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHLA
2647  007022  062737  000020  001122        ADD     #RHLA,$BDADR    ;ADDRESS OF RHLA FOR TYPEOUT
2648  007030  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHLA
2649  007034  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2650  007036  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2651  007040  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER2
2652  007044  062737  000040  001122        ADD     #RHER2,$BDADR   ;ADDRESS OF RHER2 FOR TYPEOUT
2653  007052  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER2
2654  007056  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2655  007060  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2656  007062  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHOF
2657  007066  062737  000032  001122        ADD     #RHOF,$BDADR    ;ADDRESS OF RHOF FOR TYPEOUT
2658  007074  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHOF
2659  007100  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2660  007102  104006                        ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2661  007104  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHCA
2662  007110  062737  000034  001122        ADD     #RHCA,$BDADR    ;ADDRESS OF RHCA FOR TYPEOUT
2663  007116  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHCA
2664  007122  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2665  007124  104006                        ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2666  007126  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHCC
2667  007132  062737  000036  001122        ADD     #RHCC,$BDADR    ;ADDRESS OF RHCC FOR TYPEOUT
2668  007140  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHCC
2669  007144  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2670  007146  104006                        ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2671  007150  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHSN
2672  007154  062737  000030  001122        ADD     #RHSN,$BDADR    ;ADDRESS OF RHSN FOR TYPEOUT
2673  007162  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHSN
2674  007166  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2675  007170  104006                        ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2676  007172  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHER3
2677  007176  062737  000042  001122        ADD     #RHER3,$BDADR   ;ADDRESS OF RHER3 FOR TYPEOUT
2678  007204  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHER3
2679  007210  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2680  007212  104006                        ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2681  007214  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHEC1
```

```
2692  007220  062737  000044  001122          ADD     #RHEC1,$BDADR   ;ADDRESS OF RHEC1 FOR TYPEOUT
2693  007226  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHEC1
2694  007232  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2695  007234  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2696  007236  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RHEC2
2697  007242  062737  000046  001122          ADD     #RHEC2,$BDADR   ;ADDRESS OF RHEC2 FOR TYPEOUT
2698  007250  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RHEC2
2699  007254  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2700  007256  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2701  007260                          5$:
2702
2703                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2704
2705  007260  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
2706  007264  001412                          BEQ     TST4            ;BR IF NOT
2707  007266  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
2708  007274  001406                          BEQ     TST4            ;BR IF NOT
2709  007276  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
2710  007302  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
2711  007306  000177  171576                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
2712
2713
2714                                          ;*****************************************************************
2715                                          ;*TEST 4        PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
2716                                          ;*
2717                                          ;*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
2718                                          ;*      FOR THE PORT UNDER TEST.
2719                                          ;*
2720                                          ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
2721                                          ;*      DRIVE WAS SEIZIED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
2722                                          ;*
2723                                          ;*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
2724                                          ;*      'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
2725                                          ;*      FOR PORT 'B'.  (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
2726                                          ;*      'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
2727                                          ;*
2728                                          ;*  C.  STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
2729                                          ;*      THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
2730                                          ;*
2731                                          ;*****************************************************************
2732  007312                          TST4:
2733  007312  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
2734  007314  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2735  007320  001406                          BEQ     2$              ;BR IF NOT
2736  007322  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
2737  007324  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
2738  007330  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
2739  007336  112737  000004  001102  2$:     MOVB    #4,$TSTNM       ;TEST NUMBER
2740  007344  012737  007366  001106          MOV     #TEST4,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2741  007352  012737  007366  001110          MOV     #TEST4,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
2742  007360  012737  000001  001170          MOV     #1,$TIMES       ;;DO 1 ITERATION
2743
2744
2745                                          ;;*************************************************************************
2746                                          ;END OF 'SCOPE' SETUP - START OF MAIN TEST
2747
```

```
2738  007366                                TEST4:
2739  007366  113760  001216  000010              MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2740  007374  013737  001216  001226              MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2741
2742                                        ;;**********************************************************************
2743                                        ;START THE TIMER
2744
2745  007402  005037  001244                       CLR     TIME             ;CLEAR THE ELAPSED TIME COUNTER
2746  007406  012737  003720  001246              MOV     #2000.,WATCH     ;SET WATCH TO 2000 MS
2747  007414  013737  001216  001230              MOV     PORTA,SEIZPT     ;'SEIZED' PORT ADDRESS
2748
2749                                        ;;**********************************************************************
2750                                        ;ISSUE DRIVE CLEAR COMMAND
2751
2752  007422  012760  000011  000000              MOV     #11,RHCS1(R0)    ;ISSUE A DRIVE CLEAR
2753
2754                                        ;;**********************************************************************
2755                                        ;VERIFY THAT DRIVE SEIZED BY PORT A.
2756
2757  007430  113760  001220  000010              MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2758  007436  013737  001220  001226              MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2759  007444  005037  001236                      CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2760  007450  016037  000012  001126              MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2761  007456  012737  000012  001122              MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2762  007464  060037  001122                      ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2763  007470  005037  001124                      CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
2764  007474  023737  001124  001126              CMP     $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
2765  007502  001403                              BEQ     64$              ;BR IF OK
2766  007504  104012                              ERROR   12               ;TYPE MESSAGE 12
2767  007506  005137  001236                      COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2768  007512  000240                        64$:  NOP
2769  007514  113760  001216  000010              MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
2770  007522  013737  001216  001226              MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2771  007530  005037  001236                      CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2772  007534  016037  000012  001126              MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2773  007542  012737  000012  001122              MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2774  007550  060037  001122                      ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2775  007554  012737  011600  001124              MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2776  007562  013737  001126  001156              MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2777  007570  042737  106177  001156              BIC     #^C71600,$TMP0   ;SAVE SPECIFIED BITS
2778  007576  023737  001124  001156              CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
2779  007604  001414                              BEQ     65$              ;BR IF OK
2780  007606  013737  001126  001166              MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
2781  007614  042737  071600  001166              BIC     #71600,$TMP4     ;CLEAR THE MASKED BITS
2782  007622  053737  001166  001124              BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
2783  007630  104010                              ERROR   10               ;REPORT THE ERROR
2784  007632  005137  001236                      COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2785  007636  000240                        65$:  NOP
2786  007640  005037  001236                      CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2787  007644  016037  000000  001126              MOV     RHCS1(R0),$BDDAT  ;GET CONTENTS OF RHCS1
2788  007652  012737  000000  001122              MOV     #RHCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2789  007660  060037  001122                      ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2790  007664  012737  004210  001124              MOV     #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
2791  007672  013737  001126  001156              MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2792  007700  042737  100000  001156              BIC     #^C77777,$TMP0   ;SAVE SPECIFIED BITS
2793  007706  023737  001124  001156              CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
```

```
2794  007714  001414                          BEQ     66$             ;BR IF OK
2795  007716  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2796  007724  042737  077777  001166          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
2797  007732  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2798  007740  104010                          ERROR   10              ;REPORT THE ERROR
2799  007742  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2800  007746  000240                  66$:    NOP
2801
2802                                  ;;**********************************************************************
2803                                  ;ISSUE READIN PRESET COMMAND AND SET FMT22
2804
2805  007750  012760  000023  000000          MOV     #23,RHCS1(R0)   ;ISSUE A READIN PRESET
2806  007756  012760  010000  000032          MOV     #FMT22,RHOF(R0) ;SET FMT22
2807
2808                                  ;;**********************************************************************
2809                                  ;VERIFY THAT THE DRIVE STATUS IS CORRECT
2810
2811  007764  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2812  007770  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2813  007776  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2814  010004  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2815  010010  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
2816  010016  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2817  010024  042737  106077  001156          BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
2818  010032  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2819  010040  001414                          BEQ     67$             ;BR IF OK
2820  010042  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2821  010050  042737  071700  001166          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
2822  010056  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2823  010064  104013                          ERROR   13              ;TYPE MESSAGE 13
2824  010066  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2825  010072  000240                  67$:    NOP
2826  010074  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2827  010102  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2828
2829                                  ;;**********************************************************************
2830                                  ;WAIT FOR TIMEOUT TO RELEASE DRIVE
2831
2832  010110  005760  000012          1$:     TST     RHDS1(R0)       ;WAIT FOR THE PORT TO TIME OUT
2833  010114  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
2834  010116  005737  001246                  TST     WATCH           ;CHECK THE WATCH
2835  010122  001372                          BNE     1$              ;BR IF NOT ZERO
2836  010124  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
2837  010126  000137  010444                  JMP     3$              ;BYPASS ATTN REGISTER CHECK
2838
2839                                  ;;**********************************************************************
2840                                  ;SEE IF DRIVE RETURNED TO NEUTRAL
2841
2842  010132                          2$:
2843
2844                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2845
2846  010132  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
2847  010136  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
2848  010144  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
2849  010150  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
```

```
2850  010156  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
2851  010164  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2852  010172  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
2853  010200  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2854  010206  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
2855  010214  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2856  010222  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
2857  010230  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2858  010236  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2859  010244  001006                          BNE     68$             ;BR IF NOT
2860  010246  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2861  010252  001037                          BNE     70$             ;BR IF NOT
2862  010254  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2863  010256  000137  010442                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
2864  010262  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2865  010270  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2866  010276  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
2867  010304  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2868  010310  001414                          BEQ     69$             ;BR IF ZERO
2869  010312  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2870  010320  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2871  010326  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
2872  010334  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2873  010340  001004                          BNE     70$             ;BR IF NOT
2874  010342  012737  177777  001242  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2875  010350  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
2876  010352  013737  001162  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
2877  010360  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
2878  010366  042737  100100  001162          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2879  010374  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
2880  010402  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
2881  010404  104007                          ERROR   7               ;REPORT ERROR
2882  010406  013737  001164  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
2883  010414  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
2884  010422  042737  100100  001164          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2885  010430  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2886  010436  001401                          BEQ     72$             ;BR IF OK
2887  010440  104007                          ERROR   7               ;REPORT ERROR
2888  010442  000240                  72$:    NOP
2889  010444                          3$:
2890
2891                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2892
2893  010444  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
2894  010450  001412                          BEQ     TST5            ;;BR IF NOT
2895  010452  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
2896  010460  001406                          BEQ     TST5            ;;BR IF NOT
2897  010462  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
2898  010466  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
2899  010472  000177  170412                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
2900
2901                          ;*****************************************************************
2902                          ;*TEST 5        PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'
2903                          ;*
2904                          ;*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
2905                          ;*      FOR THE PORT UNDER TEST.
```

```
                                        ;*
                                        ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
                                        ;*      DRIVE WAS SEIZIED 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.
                                        ;*
                                        ;*****************************************************************
010476                                  TST5:
010476  000004                                SCOPE                   ;INITIALIZE THE SCOPE HANDLER
010500  005737  001266                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
010504  001406                              BEQ     2$              ;BR IF NOT
010506  100002                              BPL     1$              ;BR IF JUST ENTERED TEST
010510  000137  002410                      JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
010514  012737  177777  001266    1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
010522  112737  000005  001102    2$:       MOVB    #5,$TSTNM       ;TEST NUMBER
010530  012737  010552  001106              MOV     #TEST5,$LPADR   ;LOAD LOOP ON TEST ADDRESS
010536  012737  010552  001110              MOV     #TEST5,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
010544  012737  000001  001170              MOV     #1,$TIMES       ;;DO 1 ITERATION

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

010552                                  TEST5:
010552  113760  001220  000010              MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
010560  013737  001220  001226              MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

010566  005037  001244                      CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
010572  012737  003720  001246              MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
010600  013737  001220  001230              MOV     PORTB,$EIZPT    ;'SEIZED' PORT ADDRESS

                                        ;;*************************************************************************
                                        ;ISSUE DRIVE CLEAR COMMAND

010606  012760  000011  000000              MOV     #11,RHCS1(R0)   ;ISSUE A DRIVE CLEAR

                                        ;;*************************************************************************
                                        ;VERIFY THAT DRIVE SEIZED BY PORT B.

010614  113760  001216  000010              MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
010622  013737  001216  001226              MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010630  005037  001236                      CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010634  016037  000012  001126              MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
010642  012737  000012  001122              MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010650  060037  001122                      ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
010654  005037  001124                      CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
010660  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
010666  001403                              BEQ     64$             ;BR IF OK
010670  104012                              ERROR   12              ;TYPE MESSAGE 12
```

```
2962  010672  005137  001236                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2963  010676  000240                64$:    NOP
2964  010700  113760  001220  000010        MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
2965  010706  013737  001220  001226        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2966  010714  005037  001236                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2967  010720  016037  000012  001126        MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
2968  010726  012737  000012  001122        MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2969  010734  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2970  010740  012737  011600  001124        MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2971  010746  013737  001126  001156        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2972  010754  042737  106177  001156        BIC     #^C71600,$TMP0  ;SAVE SPECIFIED BITS
2973  010762  023737  001124  001156        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2974  010770  001414                        BEQ     65$             ;BR IF OK
2975  010772  013737  001126  001166        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2976  011000  042737  071600  001166        BIC     #71600,$TMP4    ;CLEAR THE MASKED BITS
2977  011006  053737  001166  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2978  011014  104010                        ERROR   10              ;REPORT THE ERROR
2979  011016  005137  001236                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2980  011022  000240                65$:    NOP
2981  011024  005037  001236                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2982  011030  016037  000000  001126        MOV     RHCS1(R0),$BDDAT  ;GET CONTENTS OF RHCS1
2983  011036  012737  000000  001122        MOV     #RHCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2984  011044  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2985  011050  012737  004210  001124        MOV     #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
2986  011056  013737  001126  001156        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2987  011064  042737  100000  001156        BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
2988  011072  023737  001124  001156        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2989  011100  001414                        BEQ     66$             ;BR IF OK
2990  011102  013737  001126  001166        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2991  011110  042737  077777  001166        BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
2992  011116  053737  001166  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2993  011124  104010                        ERROR   10              ;REPORT THE ERROR
2994  011126  005137  001236                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2995  011132  000240                66$:    NOP
2996
2997                                  ;;*************************************************************************
2998                                  ;ISSUE READIN PRESET COMMAND AND SET FMT22
2999
3000  011134  012760  000023  000000        MOV     #23,RHCS1(R0)  ;ISSUE A READIN PRESET
3001  011142  012760  010000  000032        MOV     #FMT22,RHOF(R0) ;SET FMT22
3002
3003                                  ;;*************************************************************************
3004                                  ;VERIFY THAT THE DRIVE STATUS IS CORRECT
3005
3006  011150  005037  001236                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3007  011154  016037  000012  001126        MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3008  011162  012737  000012  001122        MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3009  011170  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3010  011174  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3011  011202  013737  001126  001156        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3012  011210  042737  106077  001156        BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
3013  011216  023737  001124  001156        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3014  011224  001414                        BEQ     67$             ;BR IF OK
3015  011226  013737  001126  001166        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3016  011234  042737  071700  001166        BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
3017  011242  053737  001166  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
3018  011250  104013                          ERROR   13              ;TYPE MESSAGE 13
3019  011252  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3020  011256  000240                  67$:    NOP
3021  011260  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
3022  011266  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3023
3024                                  ;;**********************************************************************
3025                                  ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3026
3027  011274  005760  000012          1$:     TST     RHDS1(R0)       ;WAIT FOR THE PORT TO TIME OUT
3028  011300  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
3029  011302  005737  001246                  TST     WATCH           ;CHECK THE WATCH
3030  011306  001372                          BNE     1$              ;BR IF NOT ZERO
3031  011310  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3032  011312  000137  011630                  JMP     3$              ;BYPASS ATTN REGISTER CHECK
3033
3034                                  ;;**********************************************************************
3035                                  ;SEE IF DRIVE RETURNED TO NEUTRAL
3036
3037  011316                          2$:
3038
3039                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3040
3041  011316  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3042  011322  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3043  011330  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3044  011334  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3045  011342  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3046  011350  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3047  011356  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3048  011364  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3049  011372  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3050  011400  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3051  011406  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3052  011414  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3053  011422  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3054  011430  001006                          BNE     68$             ;BR IF NOT
3055  011432  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3056  011436  001037                          BNE     70$             ;BR IF NOT
3057  011440  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3058  011442  000137  011626                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
3059  011446  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3060  011454  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3061  011462  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3062  011470  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3063  011474  001414                          BEQ     69$             ;BR IF ZERO
3064  011476  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3065  011504  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3066  011512  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3067  011520  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3068  011524  001004                          BNE     70$             ;BR IF NOT
3069  011526  012737  177777  001242  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3070  011534  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3071  011536  013737  001162  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3072  011544  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3073  011552  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
```

```
3074  011560  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3075  011566  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
3076  011570  104007                          ERROR   7               ;REPORT ERROR
3077  011572  013737  001164  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3078  011600  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3079  011606  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3080  011614  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3081  011622  001401                          BEQ     72$             ;BR IF OK
3082  011624  104007                          ERROR   7               ;REPORT ERROR
3083  011626  000240                  72$:    NOP
3084  011630                          3$:
3085
3086                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3087
3088  011630  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
3089  011634  001412                          BEQ     TST6            ;;BR IF NOT
3090  011636  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
3091  011644  001406                          BEQ     TST6            ;;BR IF NOT
3092  011646  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
3093  011652  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
3094  011656  000177  167226                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
3095
3096
3097                                  ;*****************************************************************
3098                                  ;*TEST 6        TEST RELEASE, DRIVE SEIZED BY PORT 'A'
3099                                  ;*
3100                                  ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
3101                                  ;*
3102                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
3103                                  ;*
3104                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT'A'.  VERIFY THAT THE DRIVE
3105                                  ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
3106                                  ;*      DRIVE.
3107                                  ;*
3108                                  ;*****************************************************************
3109  011662                          TST6:
3110  011662  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
3111  011664  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3112  011670  001406                          BEQ     2$              ;BR IF NOT
3113  011672  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3114  011674  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3115  011700  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3116  011706  112737  000006  001102  2$:     MOVB    #6,$TSTNM       ;TEST NUMBER
3117  011714  012737  011736  001106          MOV     #TEST6,$LPADR   ;LOAD LOOP ON TEST ADDRESS
3118  011722  012737  011736  001110          MOV     #TEST6,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
3119  011730  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
3120
3121
3122                                  ;;*************************************************************************
3123                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
3124
3125  011736                          TEST6:
3126
3127                                  ;;*************************************************************************
3128                                  ;START THE TIMER
3129
```

```
3130  011736  005037  001244                CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
3131  011742  012737  003720  001246        MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
3132
3133                                  ;;*************************************************************************
3134
3135                                  ;SEIZE THE DRIVE THROUGH PORT A
3136
3137  011750  113760  001216  000010        MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
3138  011756  013737  001216  001230        MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
3139  011764  005060  000012                CLR     RHDS1(R0)       ;WRITE RHDS1
3140  011770  013737  001220  001232        MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
3141
3142                                  ;;*************************************************************************
3143
3144                                  ;RELEASE THE DRIVE FROM PORT A
3145
3146  011776  113760  001216  000010        MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3147  012004  013737  001216  001226        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3148  012012  012760  000013  000000        MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
3149
3150                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3151
3152  012020  005037  001242                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3153  012024  012737  000012  001122        MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3154  012032  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3155  012036  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3156  012044  113760  001216  000010        MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3157  012052  016037  000012  001162        MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3158  012060  013737  001162  001156        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3159  012066  042737  100100  001156        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3160  012074  113760  001220  000010        MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3161  012102  016037  000012  001164        MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3162  012110  013737  001164  001160        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3163  012116  042737  100100  001160        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3164  012124  023737  001156  001160        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3165  012132  001006                        BNE     64$             ;BR IF NOT
3166  012134  005737  001156                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3167  012140  001037                        BNE     66$             ;BR IF NOT
3168  012142  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3169  012144  000137  012330                JMP     68$             ;BYPASS THE REST OF THE CHECKS
3170  012150  013737  001162  001126  64$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERRCR MESSAGE
3171  012156  013737  001220  001226        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3172  012164  113760  001220  000010        MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3173  012172  005737  001156                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3174  012176  001414                        BEQ     65$             ;BR IF ZERO
3175  012200  013737  001216  001226        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3176  012206  013737  001164  001126        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3177  012214  113760  001216  000010        MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3178  012222  005737  001160                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3179  012226  001004                        BNE     66$             ;BR IF NOT
3180  012230  012737  177777  001242  65$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3181  012236  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
3182  012240  013737  001162  001126  66$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3183  012246  013737  001216  001226        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3184  012254  042737  100000  001162        BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3185  012262  023737  001124  001162        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
```

```
3186  012270  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3187  012272  104007                          ERROR   7               ;REPORT ERROR
3188  012274  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3189  012302  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3190  012310  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3191  012316  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3192  012324  001401                          BEQ     68$             ;BR IF OK
3193  012326  104007                          ERROR   7               ;REPORT ERROR
3194  012330  000240                  68$:    NOP
3195  012332  005737  001242                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
3196  012336  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
3197  012340  000137  012614                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
3198  012344  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3199  012352  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3200  012360  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3201  012364  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3202  012372  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3203  012400  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3204  012404  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3205  012410  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3206  012416  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
3207  012424  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3208  012432  001414                          BEQ     69$             ;BR IF OK
3209  012434  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3210  012442  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3211  012450  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3212  012456  104017                          ERROR   17              ;TYPE MESSAGE 17
3213  012460  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3214  012464  000240                  69$:    NOP
3215  012466  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3216  012474  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3217  012502  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3218  012506  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3219  012514  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3220  012522  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3221  012526  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3222  012532  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3223  012540  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
3224  012546  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3225  012554  001414                          BEQ     70$             ;BR IF OK
3226  012556  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3227  012564  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3228  012572  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3229  012600  104017                          ERROR   17              ;TYPE MESSAGE 17
3230  012602  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3231  012606  000240                  70$:    NOP
3232  012610  000137  012646                  JMP     2$              ;GO CHECK FOR LOOP ON ERROR
3233
3234                                  ;;*****************************************************************************
3235                                  ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3236                                  ;TO RELEASE THE DRIVE
3237
3238  012614                  1$:
3239  012614  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3240  012622  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3241  012630  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR TIMEOUT TO RELEASE DRIVE
```

```
3242  012634  001004                          BNE     2$              ;BR WHEN DRIVE RELEASED
3243  012636  005737  001246                  TST     WATCH           ;CHECK THE WATCH
3244  012642  001364                          BNE     1$              ;BR IF NOT ZERO
3245  012644  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3246  012646                          2$:
3247
3248                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3249
3250  012646  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
3251  012652  001412                          BEQ     TST7            ;;BR IF NOT
3252  012654  032737  001000  177570          BIT     #SWD9,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
3253  012662  001406                          BEQ     TST7            ;;BR IF NOT
3254  012664  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
3255  012670  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
3256  012674  000177  166210                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
3257
3258                                  ;*******************************************************************
3259                                  ;*TEST 7         TEST RELEASE, DRIVE SEIZED BY PORT 'B'
3260                                  ;*
3261                                  ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
3262                                  ;*
3263                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
3264                                  ;*
3265                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
3266                                  ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
3267                                  ;*      DRIVE.
3268                                  ;*
3269                                  ;*******************************************************************
3270  012700                          TST7:
3271  012700  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
3272  012702  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3273  012706  001406                          BEQ     2$              ;BR IF NOT
3274  012710  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3275  012712  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3276  012716  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3277  012724  112737  000007  001102  2$:     MOVB    #7,$TSTNM       ;TEST NUMBER
3278  012732  012737  012754  001106          MOV     #TEST7,$LPADR   ;LOAD LOOP ON TEST ADDRESS
3279  012740  012737  012754  001110          MOV     #TEST7,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
3280  012746  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
3281
3282
3283                                  ;;*******************************************************************
3284                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
3285
3286  012754                          TEST7:
3287
3288                                  ;;*******************************************************************
3289                                  ;START THE TIMER
3290
3291  012754  005037  001244                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
3292  012760  012737  003720  001246          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
3293
3294                                  ;;*******************************************************************
3295
3296                                          ;SEIZE THE DRIVE THROUGH PORT B
3297
```

```
3298  012766  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
3299  012774  013737  001220  001230          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
3300  013002  005060  000012                  CLR     RHDS1(R0)     ;WRITE RHDS1
3301  013006  013737  001216  001232          MOV     PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
3302
3303                                  ;;*************************************************************************
3304
3305                                  ;RELEASE THE DRIVE FROM PORT B
3306
3307  013014  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3308  013022  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3309  013030  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
3310
3311                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3312
3313  013036  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3314  013042  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3315  013050  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3316  013054  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3317  013062  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3318  013070  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3319  013076  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3320  013104  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3321  013112  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3322  013120  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3323  013126  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3324  013134  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3325  013142  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3326  013150  001006                          BNE     64$             ;BR IF NOT
3327  013152  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3328  013156  001037                          BNE     66$             ;BR IF NOT
3329  013160  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3330  013162  000137  013346                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
3331  013166  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3332  013174  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3333  013202  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3334  013210  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3335  013214  001414                          BEQ     65$             ;BR IF ZERO
3336  013216  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3337  013224  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3338  013232  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3339  013240  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3340  013244  001004                          BNE     66$             ;BR IF NOT
3341  013246  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3342  013254  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3343  013256  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3344  013264  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3345  013272  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3346  013300  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3347  013306  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3348  013310  104007                          ERROR   7               ;REPORT ERROR
3349  013312  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3350  013320  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3351  013326  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3352  013334  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3353  013342  001401                          BEQ     68$             ;BR IF OK
```

```
3354  013344  104007                          ERROR   7               ;REPORT ERROR
3355  013346  000240                  68$:    NOP
3356  013350  005737  001242                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
3357  013354  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
3358  013356  000137  013632                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
3359  013362  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3360  013370  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3361  013376  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3362  013402  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3363  013410  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3364  013416  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3365  013422  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3366  013426  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3367  013434  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
3368  013442  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3369  013450  001414                          BEQ     69$             ;BR IF OK
3370  013452  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3371  013460  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3372  013466  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3373  013474  104017                          ERROR   17              ;TYPE MESSAGE 17
3374  013476  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3375  013502  000240                  69$:    NOP
3376  013504  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3377  013512  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3378  013520  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3379  013524  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3380  013532  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3381  013540  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3382  013544  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3383  013550  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3384  013556  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
3385  013564  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3386  013572  001414                          BEQ     70$             ;BR IF OK
3387  013574  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3388  013602  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3389  013610  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3390  013616  104017                          ERROR   17              ;TYPE MESSAGE 17
3391  013620  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3392  013624  000240                  70$:    NOP
3393  013626  000137  013664                  JMP     2$              ;GO CHECK FOR LOOP ON ERROR
3394
3395                                  ;;*****************************************************************
3396                                  ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3397                                  ;TO RELEASE THE DRIVE
3398
3399  013632                          1$:
3400  013632  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3401  013640  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3402  013646  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3403  013652  001004                          BNE     2$              ;BR WHEN DRIVE RELEASED
3404  013654  005737  001246                  TST     WATCH           ;CHECK THE WATCH
3405  013660  001364                          BNE     1$              ;BR IF NOT ZERO
3406  013662  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3407  013664                          2$:
3408
3409                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
```

```
                                                TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
                                                BEQ     TST10           ;;BR IF NOT
                                                BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
                                                BEQ     TST10           ;;BR IF NOT
                                                CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
                                                CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
                                                JMP     @$LPERR         ;GO TO THE LOOP ADDRESS

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


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

013772                                  TEST10:
013772  113760  001216  000010                  MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
014000  013737  001216  001226                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014006  013737  001216  001230                  MOV     PORTA,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE

                                        ;;*********************************************************************
                                        ;ISSUE A RELEASE COMMAND

014014  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND

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


                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

014022  005037  001242                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
014026  012737  000012  001122                  MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
014034  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
014040  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
014046  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
```

```
3466  014054  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3467  014062  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3468  014070  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3469  014076  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3470  014104  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3471  014112  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3472  014120  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3473  014126  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3474  014134  001006                          BNE     64$             ;BR IF NOT
3475  014136  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3476  014142  001037                          BNE     66$             ;BR IF NOT
3477  014144  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3478  014146  000137  014332                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
3479  014152  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3480  014160  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3481  014166  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3482  014174  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3483  014200  001414                          BEQ     65$             ;BR IF ZERO
3484  014202  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3485  014210  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3486  014216  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3487  014224  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3488  014230  001004                          BNE     66$             ;BR IF NOT
3489  014232  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3490  014240  104030                          ERROR   30              ;TYPE ERROR MESSAGE 30
3491  014242  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3492  014250  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3493  014256  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3494  014264  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3495  014272  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3496  014274  104007                          ERROR   7               ;REPORT ERROR
3497  014276  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3498  014304  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3499  014312  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3500  014320  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3501  014326  001401                          BEQ     68$             ;BR IF OK
3502  014330  104007                          ERROR   7               ;REPORT ERROR
3503  014332  000240                  68$:    NOP
3504
3505                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3506
3507  014334  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
3508  014340  001412                          BEQ     TST11           ;;BR IF NOT
3509  014342  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
3510  014350  001406                          BEQ     TST11           ;;BR IF NOT
3511  014352  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
3512  014356  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
3513  014362  000177  164522                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
3514
3515                          ;*****************************************************************
3516                          ;*TEST 11       TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
3517                          ;*
3518                          ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
3519                          ;*
3520                          ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
3521                          ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
```

```
3522                                    ;*
3523                                    ;*********************************************************************
3524  014366                            TST11:
3525  014366  000004                           SCOPE                   ;INITIALIZE THE SCOPE HANDLER
3526  014370  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3527  014374  001406                          BEQ     2$              ;BR IF NOT
3528  014376  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3529  014400  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3530  014404  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3531  014412  112737  000011  001102  2$:     MOVB    #11,$TSTNM      ;TEST NUMBER
3532  014420  012737  014442  001106          MOV     #TEST11,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3533  014426  012737  014442  001110          MOV     #TEST11,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3534  014434  012737  000144  001170          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3535
3536                                    ;;*********************************************************************
3537                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
3538
3539  014442                            TEST11:
3540  014442  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3541  014450  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3542  014456  013737  001220  001230          MOV     PORTB,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3543
3544                                    ;;*********************************************************************
3545                                    ;ISSUE A RELEASE COMMAND
3546
3547  014464  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND
3548
3549                                    ;;*********************************************************************
3550                                    ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
3551
3552
3553                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3554
3555  014472  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3556  014476  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3557  014504  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3558  014510  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3559  014516  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3560  014524  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3561  014532  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3562  014540  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3563  014546  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3564  014554  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3565  014562  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3566  014570  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3567  014576  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3568  014604  001006                          BNE     64$             ;BR IF NOT
3569  014606  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3570  014612  001037                          BNE     66$             ;BR IF NOT
3571  014614  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3572  014616  000137  015002                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
3573  014622  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3574  014630  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3575  014636  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3576  014644  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3577  014650  001414                          BEQ     65$             ;BR IF ZERO
```

```
3578  014652  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3579  014660  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3580  014666  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3581  014674  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3582  014700  001004                          BNE     66$             ;BR IF NOT
3583  014702  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3584  014710  104030                          ERROR   30              ;TYPE ERROR MESSAGE 30
3585  014712  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3586  014720  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3587  014726  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3588  014734  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3589  014742  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3590  014744  104007                          ERROR   7               ;REPORT ERROR
3591  014746  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3592  014754  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3593  014762  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3594  014770  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3595  014776  001401                          BEQ     68$             ;BR IF OK
3596  015000  104007                          ERROR   7               ;REPORT ERROR
3597  015002  000240                  68$:    NOP
3598
3599                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3600
3601  015004  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
3602  015010  001412                          BEQ     TST12           ;;BR IF NOT
3603  015012  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
3604  015020  001406                          BEQ     TST12           ;;BR IF NOT
3605  015022  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
3606  015026  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
3607  015032  000177  164052                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
3608
3609
3610                                          ;*****************************************************************
3611                                          ;*TEST 12       TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
3612                                          ;*
3613                                          ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3614                                          ;*      PORT TO RELEASE THE DRIVE.
3615                                          ;*
3616                                          ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RHDS1 THROUGH PORT 'A'.
3617                                          ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
3618                                          ;*
3619                                          ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
3620                                          ;*      DOES NOT RETURN TO NEUTRAL.
3621                                          ;*
3622                                          ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
3623                                          ;*      DOES NOT RETURN TO NEUTRAL.
3624                                          ;*
3625                                          ;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
3626                                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3627                                          ;*
3628                                          ;*****************************************************************
3629  015036                          TST12:
3630  015036  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
3631  015040  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3632  015044  001406                          BEQ     2$              ;BR IF NOT
3633  015046  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
```

```
3634  015050  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3635  015054  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3636  015062  112737  000012  001102  2$:     MOVB    #12,$TSTNM      ;TEST NUMBER
3637  015070  012737  015112  001106          MOV     #TEST12,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3638  015076  012737  015112  001110          MOV     #TEST12,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3639  015104  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
3640
3641
3642                                  ;;**********************************************************************
3643                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
3644
3645  015112                          TEST12:
3646
3647                                  ;;**********************************************************************
3648
3649
3650
3651  015112  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
3652  015120  013737  001216  001230          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
3653  015126  005060  000012                  CLR     RHDS1(R0)       ;WRITE RHDS1
3654  015132  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
3655  015140  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3656  015146  013737  001220  001232          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
3657  015154  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
3658  015162  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
3659  015166  062737  000012  001122          ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
3660  015174  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
3661  015200  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
3662  015206  001403                          BEQ     .+10            ;BR IF IT IS
3663  015210  104004                          ERROR   4               ;REPORT THE ERROR
3664  015212  000137  016406                  JMP     1$              ;BYPASS REST OF THE SUBTEST
3665  015216  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
3666  015224  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3667  015232  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
3668  015240  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
3669  015246  013737  001124  001160          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
3670  015254  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
3671  015260  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
3672  015266  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
3673  015274  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
3674  015302  001401                          BEQ     .+4             ;BR IF THEY ARE
3675  015304  104005                          ERROR 5                 ;REPORT THE ERROR
3676
3677                                  ;;**********************************************************************
3678                                  ;DRIVE CLEAR THROUGH PORT A FIRST
3679
3680  015306  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT A
3681
3682                                  ;;**********************************************************************
3683                                  ;VERIFY THAT DRIVE STILL SEIZED BY PORT A
3684
3685  015314  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
3686  015322  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3687  015330  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3688  015334  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;GET CONTENTS OF RHDS1
3689  015342  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
```

```
3690  015350  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3691  015354  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3692  015360  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3693  015366  042737  100000  001156            BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3694  015374  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3695  015402  001414                            BEQ     64$             ;BR IF OK
3696  015404  013737  001126  001166            MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3697  015412  042737  077777  001166            BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3698  015420  053737  001166  001124            BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3699  015426  104033                            ERROR   33              ;TYPE MESSAGE 33
3700  015430  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3701  015434  000240                    64$:    NOP
3702  015436  113760  001216  000010            MOVB    PORTA,RHCS2(R0)   ;SELECT PORT A
3703  015444  013737  001216  001226            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3704  015452  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3705  015456  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3706  015464  012737  000012  001122            MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3707  015472  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3708  015476  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3709  015504  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3710  015512  042737  100000  001156            BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3711  015520  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3712  015526  001414                            BEQ     65$             ;BR IF OK
3713  015530  013737  001126  001166            MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3714  015536  042737  077777  001166            BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3715  015544  053737  001166  001124            BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3716  015552  104033                            ERROR   33              ;TYPE MESSAGE 33
3717  015554  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3718  015560  000240                    65$:    NOP
3719
3720                                    ;;**********************************************************************
3721                                    ;NOW ISSUE MASSBUS INIT
3722
3723  015562  012760  000040  000010            MOV     #CLR,RHCS2(R0)  ;ISSUE MASSBUS INIT
3724
3725                                    ;;**********************************************************************
3726                                    ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
3727
3728  015570  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3729  015576  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3730  015604  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3731  015610  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3732  015616  012737  000012  001122            MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3733  015624  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3734  015630  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3735  015634  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3736  015642  042737  100000  001156            BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3737  015650  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3738  015656  001414                            BEQ     66$             ;BR IF OK
3739  015660  013737  001126  001166            MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3740  015666  042737  077777  001166            BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3741  015674  053737  001166  001124            BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3742  015702  104034                            ERROR   34              ;TYPE MESSAGE 34
3743  015704  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3744  015710  000240                    66$:    NOP
3745  015712  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
```

```
3746  015720  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3747  015726  005037  001236                  CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
3748  015732  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3749  015740  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3750  015746  060037  001122                  ADD     R0,$BDADR    ;ADD RH11 BASE ADDRESS
3751  015752  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3752  015760  013737  001126  001156          MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3753  015766  042737  100000  001156          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3754  015774  023737  001124  001156          CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
3755  016002  001414                          BEQ     67$          ;BR IF OK
3756  016004  013737  001126  001166          MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
3757  016012  042737  077777  001166          BIC     #77777,$TMP4 ;CLEAR THE MASKED BITS
3758  016020  053737  001166  001124          BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3759  016026  104034                          ERROR   34           ;TYPE MESSAGE 34
3760  016030  005137  001236                  COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
3761  016034  000240                  67$:    NOP
3762
3763                                          ;RELEASE THE DRIVE FROM PORT A
3764
3765  016036  113760  001216  000C10          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3766  016044  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3767  016052  012760  000013  000000          MOV     #13,RHCS1(R0)  ;ISSUE RELEASE THROUGH PORT A
3768
3769                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3770
3771  016060  005037  001242                  CLR     RELERR       ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3772  016064  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3773  016072  060037  001122                  ADD     R0,$BDADR    ;ADD THE I/O BASE ADDRESS
3774  016076  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3775  016104  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3776  016112  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3777  016120  013737  001162  001156          MOV     $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
3778  016126  042737  100100  001156          BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3779  016134  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3780  016142  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3781  016150  013737  001164  001160          MOV     $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
3782  016156  042737  100100  001160          BIC     #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3783  016164  023737  001156  001160          CMP     $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3784  016172  001006                          BNE     68$          ;BR IF NOT
3785  016174  005737  001156                  TST     $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3786  016200  001045                          BNE     70$          ;BR IF NOT
3787  016202  104046                          ERROR   46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3788  016204  000137  016404                  JMP     72$          ;BYPASS THE REST OF THE CHECKS
3789  016210  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3790  016216  013737  001220  001226          MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3791  016224  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3792  016232  005737  001156                  TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
3793  016236  001414                          BEQ     69$          ;BR IF ZERO
3794  016240  013737  001216  001226          MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3795  016246  013737  001164  001126          MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
3796  016254  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3797  016262  005737  001160                  TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
3798  016266  001012                          BNE     70$          ;BR IF NOT
3799  016270  012737  177777  001242  69$:    MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
3800  016276  012760  000011  000000          MOV     #11,RHCS1(R0)  ;CLEAR THE DRIVE
3801  016304  012760  000013  000000          MOV     #13,RHCS1(R0)  ;RELEASE THE DRIVE
```

```
3802  016312  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3803  016314  013737  001162  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
3804  016322  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3805  016330  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3806  016336  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3807  016344  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
3808  016346  104007                          ERROR   7               ;REPORT ERROR
3809  016350  013737  001164  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
3810  016356  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3811  016364  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3812  016372  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3813  016400  001401                          BEQ     72$             ;BR IF OK
3814  016402  104007                          ERROR   7               ;REPORT ERROR
3815  016404  000240                  72$:    NOP
3816  016406                          1$:
3817
3818                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3819
3820  016406  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
3821  016412  001412                          BEQ     TST13           ;;BR IF NOT
3822  016414  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
3823  016422  001406                          BEQ     TST13           ;;BR IF NOT
3824  016424  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
3825  016430  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
3826  016434  000177  162450                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
3827
3828                                  ;*****************************************************************
3829                                  ;*TEST 13       TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
3830                                  ;*
3831                                  ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3832                                  ;*      PORT TO RELEASE THE DRIVE.
3833                                  ;*
3834                                  ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RHDS1 THROUGH PORT 'B'.
3835                                  ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
3836                                  ;*
3837                                  ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
3838                                  ;*      DOES NOT RETURN TO NEUTRAL.
3839                                  ;*
3840                                  ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
3841                                  ;*      DOES NOT RETURN TO NEUTRAL.
3842                                  ;*
3843                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
3844                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3845                                  ;*
3846                                  ;*****************************************************************
3847  016440                          TST13:
3848  016440  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
3849  016442  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3850  016446  001406                          BEQ     2$              ;BR IF NOT
3851  016450  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3852  016452  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3853  016456  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3854  016464  112737  000013  001102  2$:     MOVB    #13,$TSTNM      ;TEST NUMBER
3855  016472  012737  016514  001106          MOV     #TEST13,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3856  016500  012737  016514  001110          MOV     #TEST13,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3857  016506  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
```

```
3858
3859
3860                                    ;;*****************************************************************************
3861                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
3862
3863  016514                            TEST13:
3864
3865                                    ;;*****************************************************************************
3866
3867                                            ;SEIZE THE DRIVE THROUGH PORT B
3868
3869  016514  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
3870  016522  013737  001220  001230            MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
3871  016530  005060  000012                    CLR     RHDS1(R0)       ;WRITE RHDS1
3872  016534  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3873  016542  013737  001216  001226            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3874  016550  013737  001216  001232            MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
3875  016556  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;SEE IF DRIVE SEIZED BY PORT B
3876  016564  010037  001122                    MOV     R0,$BDADR       ;RH11 BASE ADDRESS
3877  016570  062737  000012  001122            ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
3878  016576  005037  001124                    CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
3879  016602  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
3880  016610  001403                            BEQ     .+10            ;BR IF IT IS
3881  016612  104004                            ERROR   4               ;REPORT THE ERROR
3882  016614  000137  020010                    JMP     1$              ;BYPASS REST OF THE SUBTEST
3883  016620  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3884  016626  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3885  016634  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
3886  016642  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
3887  016650  013737  001124  001160            MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
3888  016656  005137  001160                    COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
3889  016662  013737  001126  001156            MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
3890  016670  043737  001160  001156            BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
3891  016676  023737  001124  001156            CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
3892  016704  001401                            BEQ     .+4             ;BR IF THEY ARE
3893  016706  104005                            ERROR 5                 ;REPORT THE ERROR
3894
3895                                    ;;*****************************************************************************
3896                                    ;DRIVE CLEAR THROUGH PORT B FIRST
3897
3898  016710  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT B
3899
3900                                    ;;*****************************************************************************
3901                                    ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
3902
3903  016716  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3904  016724  013737  001216  001226            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3905  016732  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3906  016736  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3907  016744  012737  000012  001122            MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3908  016752  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3909  016756  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3910  016762  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3911  016770  042737  100000  001156            BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3912  016776  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3913  017004  001414                            BEQ     64$             ;BR IF OK
```

```
3914  017006  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3915  017014  042737  077777  001166          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3916  017022  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3917  017030  104033                          ERROR   33              ;TYPE MESSAGE 33
3918  017032  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3919  017036  000240                  64$:    NOP
3920  017040  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3921  017046  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3922  017054  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3923  017060  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3924  017066  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3925  017074  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3926  017100  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3927  017106  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3928  017114  042737  100000  001156          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3929  017122  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3930  017130  001414                          BEQ     65$             ;BR IF OK
3931  017132  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3932  017140  042737  077777  001166          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3933  017146  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3934  017154  104033                          ERROR   33              ;TYPE MESSAGE 33
3935  017156  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3936  017162  000240                  65$:    NOP
3937
3938                                  ;;*****************************************************************
3939                                  ;NOW ISSUE MASSBUS INIT
3940
3941  017164  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;ISSUE MASSBUS INIT
3942
3943                                  ;;*****************************************************************
3944                                  ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
3945
3946  017172  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
3947  017200  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3948  017206  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3949  017212  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3950  017220  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3951  017226  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3952  017232  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3953  017236  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3954  017244  042737  100000  001156          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3955  017252  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3956  017260  001414                          BEQ     66$             ;BR IF OK
3957  017262  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3958  017270  042737  077777  001166          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3959  017276  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3960  017304  104034                          ERROR   34              ;TYPE MESSAGE 34
3961  017306  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3962  017312  000240                  66$:    NOP
3963  017314  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3964  017322  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3965  017330  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3966  017334  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
3967  017342  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3968  017350  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3969  017354  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
```

```
3970  017362  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3971  017370  042737  100000  001156          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3972  017376  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3973  017404  001414                          BEQ     67$             ;BR IF OK
3974  017406  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3975  017414  042737  077777  001166          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3976  017422  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3977  017430  104034                          ERROR   34              ;TYPE MESSAGE 34
3978  017432  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3979  017436  000240                  67$:    NOP
3980
3981                                          ;RELEASE THE DRIVE FROM PORT B
3982
3983  017440  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
3984  017446  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3985  017454  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
3986
3987                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3988
3989  017462  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3990  017466  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
3991  017474  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3992  017500  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3993  017506  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
3994  017514  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3995  017522  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3996  017530  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3997  017536  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
3998  017544  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3999  017552  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4000  017560  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4001  017566  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4002  017574  001006                          BNE     68$             ;BR IF NOT
4003  017576  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4004  017602  001045                          BNE     70$             ;BR IF NOT
4005  017604  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4006  017606  000137  020006                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
4007  017612  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4008  017620  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4009  017626  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4010  017634  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4011  017640  001414                          BEQ     69$             ;BR IF ZERO
4012  017642  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4013  017650  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4014  017656  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4015  017664  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4016  017670  001012                          BNE     70$             ;BR IF NOT
4017  017672  012737  177777  001242  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4018  017700  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
4019  017706  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4020  017714  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4021  017716  013737  001162  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
4022  017724  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4023  017732  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
4024  017740  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4025  017746  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
```

```
4026  017750  104007                          ERROR   7               ;REPORT ERROR
4027  017752  013737  001164  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
4028  017760  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4029  017766  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
4030  017774  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4031  020002  001401                          BEQ     72$             ;BR IF OK
4032  020004  104007                          ERROR   7               ;REPORT ERROR
4033  020006  000240                  72$:    NOP
4034  020010                          1$:
4035
4036                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
4037
4038  020010  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
4039  020014  001412                          BEQ     TST14           ;;BR IF NOT
4040  020016  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
4041  020024  001406                          BEQ     TST14           ;;BR IF NOT
4042  020026  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
4043  020032  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
4044  020036  000177  161046                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
4045
4046
4047                                  ;*****************************************************************
4048                                  ;*TEST 14       TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
4049                                  ;*
4050                                  ;*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
4051                                  ;*      SEIZING PORT.
4052                                  ;*
4053                                  ;*  A.  SET EACH PORT 'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
4054                                  ;*      SET.
4055                                  ;*
4056                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
4057                                  ;*
4058                                  ;*  C.  ISSUE A MASSBUS CLEAR.
4059                                  ;*
4060                                  ;*
4061                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE ATTENTION
4062                                  ;*      BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
4063                                  ;*      'B' IS STILL SET.
4064                                  ;*
4065                                  ;*****************************************************************
4066  020042                          TST14:
4067  020042  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
4068  020044  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4069  020050  001406                          BEQ     2$              ;BR IF NOT
4070  020052  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
4071  020054  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
4072  020060  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4073  020066  112737  000014  001102  2$:     MOVB    #14,$TSTNM      ;TEST NUMBER
4074  020074  012737  020116  001106          MOV     #TEST14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4075  020102  012737  020116  001110          MOV     #TEST14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4076  020110  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
4077
4078
4079                                  ;;*****************************************************************
4080                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
4081
```

```
4082  020116                          TEST14:
4083                                  ;;*************************************************************************
4084
4085                                          ;SET ATTENTION BITS FOR BOTH PORTS
4086
4087  020116  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT 'A'
4088  020124  012760  177777  000014          MOV     #-1,RHER1(R0)   ;FORCE ERRORS
4089  020132  005060  000014                  CLR     RHER1(R0)       ;CLEAR THE ERRORS
4090  020136  013760  001220  000010          MOV     PORTB,RHCS2(R0) ;SELECT THE OTHER PORT
4091  020144  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
4092  020150  001775                          BEQ     .-4             ;BR IF DRIVE HASN'T TIMED OUT
4093  020152  012760  177777  000014          MOV     #-1,RHER1(R0)   ;FORCE ERRORS ON PORT 'B'
4094  020160  005060  000014                  CLR     RHER1(R0)       ;CLEAR THE ERRORS
4095  020164  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT "A" AGAIN
4096  020172  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
4097  020176  001775                          BEQ     .-4             ;BR IF DRIVE HASN'T TIMED OUT
4098
4099                                  ;;*************************************************************************
4100                                  ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
4101
4102  020200  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
4103  020206  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4104  020214  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4105  020220  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4106  020226  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4107  020234  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4108  020240  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4109  020246  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4110  020254  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4111  020262  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4112  020270  001414                          BEQ     64$             ;BR IF OK
4113  020272  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4114  020300  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4115  020306  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4116  020314  104010                          ERROR   10              ;REPORT THE ERROR
4117  020316  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4118  020322  000240                  64$:    NOP
4119  020324  005737  001236                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
4120  020330  001402                          BEQ     .+6             ;BR IF IT WAS
4121  020332  000137  021340                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4122  020336  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4123  020344  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4124  020352  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4125  020356  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4126  020364  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4127  020372  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4128  020376  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4129  020404  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4130  020412  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4131  020420  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4132  020426  001414                          BEQ     65$             ;BR IF OK
4133  020430  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4134  020436  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4135  020444  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4136  020452  104010                          ERROR   10              ;REPORT THE ERROR
4137  020454  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
4138  020460  000240                  65$:    NOP
4139  020462  005737  001236                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
4140  020466  001402                          BEQ     .+6             ;BR IF IT WAS
4141  020470  000137  021340                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4142
4143                                  ;;**********************************************************************
4144
4145                                          ;SEIZE THE DRIVE THROUGH PORT A
4146
4147  020474  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
4148  020502  013737  001216  001230          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
4149  020510  005060  000C12                  CLR     RHDS1(R0)       ;WRITE RHDS1
4150  020514  013737  001220  001232          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
4151
4152                                  ;;**********************************************************************
4153                                  ;ISSUE MASSBUS INIT TO PORT A
4154
4155  020522  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;MASSBUS INIT
4156  020530  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A AGAIN
4157
4158                                  ;;**********************************************************************
4159                                  ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
4160
4161  020536  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4162  020542  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;GET CONTENTS OF RHDS1
4163  020550  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4164  020556  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4165  020562  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
4166  020566  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4167  020574  042737  077777  001156          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
4168  020602  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4169  020610  001414                          BEQ     66$             ;BR IF OK
4170  020612  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4171  020620  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4172  020626  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4173  020634  104047                          ERROR   47              ;TYPE MESSAGE 47
4174  020636  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4175  020642  000240                  66$:    NOP
4176
4177                                  ;;**********************************************************************
4178
4179                                          ;RELEASE THE DRIVE FROM PORT A
4180
4181  020644  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
4182  020652  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4183  020660  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
4184
4185                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4186
4187  020666  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4188  020672  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
4189  020700  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4190  020704  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
4191  020712  113760  001215  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4192  020720  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4193  020726  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
4194  020734  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4195  020742  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4196  020750  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4197  020756  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4198  020764  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4199  020772  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4200  021000  001006                          BNE     67$             ;BR IF NOT
4201  021002  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4202  021006  001045                          BNE     69$             ;BR IF NOT
4203  021010  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4204  021012  000137  021212                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
4205  021016  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4206  021024  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4207  021032  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4208  021040  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4209  021044  001414                          BEQ     68$             ;BR IF ZERO
4210  021046  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4211  021054  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4212  021062  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4213  021070  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4214  021074  001012                          BNE     69$             ;BR IF NOT
4215  021076  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4216  021104  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
4217  021112  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4218  021120  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4219  021122  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
4220  021130  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4221  021136  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
4222  021144  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4223  021152  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
4224  021154  104007                          ERROR   7               ;REPORT ERROR
4225  021156  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
4226  021164  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4227  021172  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
4228  021200  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4229  021206  001401                          BEQ     71$             ;BR IF OK
4230  021210  104007                          ERROR   7               ;REPORT ERROR
4231  021212  000240                  71$:    NOP
4232
4233                                          ;;*************************************************************************
4234                                          ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)
4235
4236  021214  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4237  021222  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4238  021230  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4239  021234  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4240  021242  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4241  021250  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4242  021254  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4243  021262  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4244  021270  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4245  021276  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4246  021304  001414                          BEQ     72$             ;BR IF OK
4247  021306  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4248  021314  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4249  021322  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
4250  021330  104050                          ERROR   50              ;TYPE MESSAGE 50
4251  021332  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4252  021336  000240                  72$:    NOP
4253  021340                          1$:
4254
4255                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
4256
4257  021340  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
4258  021344  001412                          BEQ     TST15           ;;BR IF NOT
4259  021346  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
4260  021354  001406                          BEQ     TST15           ;;BR IF NOT
4261  021356  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
4262  021362  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
4263  021366  000177  157516                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
4264
4265                                  ;*****************************************************************
4266                                  ;*TEST 15       TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
4267                                  ;*
4268                                  ;*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
4269                                  ;*      SEIZING PORT.
4270                                  ;*
4271                                  ;*  A.  SET EACH PORT'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
4272                                  ;*      SET.
4273                                  ;*
4274                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
4275                                  ;*
4276                                  ;*  C.  ISSUE A MASSBUS CLEAR.
4277                                  ;*
4278                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE ATTENTION
4279                                  ;*      BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
4280                                  ;*      'A' IS STILL SET.
4281                                  ;*
4282                                  ;*****************************************************************
4283  021372                          TST15:
4284  021372  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
4285  021374  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4286  021400  001406                          BEQ     2$              ;BR IF NOT
4287  021402  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
4288  021404  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
4289  021410  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4290  021416  112737  000015  001102  2$:     MOVB    #15,$TSTNM      ;TEST NUMBER
4291  021424  012737  021446  001106          MOV     #TEST15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4292  021432  012737  021446  001110          MOV     #TEST15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4293  021440  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
4294
4295
4296                                  ;;*************************************************************************
4297                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
4298
4299  021446                          TEST15:
4300                                  ;;*************************************************************************
4301
4302                                          ;SET ATTENTION BITS FOR BOTH PORTS
4303
4304  021446  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT 'A'
4305  021454  012760  177777  000014          MOV     #-1,RHER1(R0)   ;FORCE ERRORS
```

```
4306  021462  005060  000014                CLR     RHER1(R0)           ;CLEAR THE ERRORS
4307  021466  013760  001220  000010        MOV     PORTB,RHCS2(R0)     ;SELECT THE OTHER PORT
4308  021474  005760  000012                TST     RHDS1(R0)           ;WAIT FOR DRIVE TO TIMEOUT
4309  021500  001775                        BEQ     .-4                 ;BR IF DRIVE HASN'T TIMED OUT
4310  021502  012760  177777  000014        MOV     #-1,RHER1(R0)       ;FORCE ERRORS ON PORT 'B'
4311  021510  005060  000014                CLR     RHER1(R0)           ;CLEAR THE ERRORS
4312  021514  113760  001216  000010        MOVB    PORTA,RHCS2(R0)     ;SELECT PORT "A" AGAIN
4313  021522  005760  000012                TST     RHDS1(R0)           ;WAIT FOR DRIVE TO TIMEOUT
4314  021526  001775                        BEQ     .-4                 ;BR IF DRIVE HASN'T TIMED OUT
4315
4316                                ;;**************************************************************************
4317                                ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
4318
4319  021530  113760  001220  000010        MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4320  021536  013737  001220  001226        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4321  021544  005037  001236                CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
4322  021550  016037  000012  001126        MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4323  021556  012737  000012  001122        MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4324  021564  060037  001122                ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
4325  021570  012737  100000  001124        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4326  021576  013737  001126  001156        MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
4327  021604  042737  077777  001156        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4328  021612  023737  001124  001156        CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
4329  021620  001414                        BEQ     64$              ;BR IF OK
4330  021622  013737  001126  001166        MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
4331  021630  042737  100000  001166        BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
4332  021636  053737  001166  001124        BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
4333  021644  104010                        ERROR   10               ;REPORT THE ERROR
4334  021646  005137  001236                COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
4335  021652  000240                64$:    NOP
4336  021654  005737  001236                TST     CKERR            ;WAS ATTN BIT FOR PORT B SET ?
4337  021660  001402                        BEQ     .+6              ;BR IF IT WAS
4338  021662  000137  022670                JMP     1$               ;BYPASS REST OF TEST IF NOT
4339  021666  113760  001216  000010        MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
4340  021674  013737  001216  001226        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4341  021702  005037  001236                CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
4342  021706  016037  000012  001126        MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4343  021714  012737  000012  001122        MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4344  021722  060037  001122                ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
4345  021726  012737  100000  001124        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4346  021734  013737  001126  001156        MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
4347  021742  042737  077777  001156        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4348  021750  023737  001124  001156        CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
4349  021756  001414                        BEQ     65$              ;BR IF OK
4350  021760  013737  001126  001166        MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
4351  021766  042737  100000  001166        BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
4352  021774  053737  001166  001124        BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
4353  022002  104010                        ERROR   10               ;REPORT THE ERROR
4354  022004  005137  001236                COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
4355  022010  000240                65$:    NOP
4356  022012  005737  001236                TST     CKERR            ;WAS ATTN BIT FOR PORT A SET ?
4357  022016  001402                        BEQ     .+6              ;BR IF IT WAS
4358  022020  000137  022670                JMP     1$               ;BYPASS REST OF TEST IF NOT
4359
4360                                ;;**************************************************************************
4361
```

```
                                        ;SEIZE THE DRIVE THROUGH PORT B
                                        ;
022024  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
022032  013737  001220  001230          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
022040  005060  000012                  CLR     RHDS1(R0)       ;WRITE RHDS1
022044  013737  001216  001232          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
                                        ;
                                        ;;*************************************************************************
                                        ;ISSUE MASSBUS INIT TO PORT B
                                        ;
022052  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;MASSBUS INIT
022060  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B AGAIN
                                        ;
                                        ;;*************************************************************************
                                        ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
                                        ;
022066  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
022072  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
022100  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022106  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
022112  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
022116  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
022124  042737  077777  001156          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
022132  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
022140  001414                          BEQ     66$             ;BR IF OK
022142  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
022150  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
022156  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
022164  104047                          ERROR   47              ;TYPE MESSAGE 47
022166  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
022172  000240                  66$:     NOP
                                        ;
                                        ;;*************************************************************************
                                        ;
                                        ;RELEASE THE DRIVE FROM PORT B
                                        ;
022174  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
022202  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022210  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
                                        ;
                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL
                                        ;
022216  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
022222  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
022230  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
022234  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
022242  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
022250  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022256  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
022264  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022272  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
022300  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022306  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
022314  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022322  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
022330  001006                          BNE     67$             ;BR IF NOT
```

```
4418  022332  005737  001156                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4419  022336  001045                            BNE     69$             ;BR IF NOT
4420  022340  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4421  022342  000137  022542                    JMP     71$             ;BYPASS THE REST OF THE CHECKS
4422  022346  013737  001162  001126  67$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4423  022354  013737  001220  001226            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4424  022362  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4425  022370  005737  001156                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4426  022374  001414                            BEQ     68$             ;BR IF ZERO
4427  022376  013737  001216  001226            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4428  022404  013737  001164  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4429  022412  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4430  022420  005737  001160                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4431  022424  001012                            BNE     69$             ;BR IF NOT
4432  022426  012737  177777  001242  68$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4433  022434  012760  000011  000000            MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
4434  022442  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4435  022450  104026                            ERROR   26              ;TYPE ERROR MESSAGE 26
4436  022452  013737  001162  001126  69$:     MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
4437  022460  013737  001216  001226            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4438  022466  042737  100000  001162            BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
4439  022474  023737  001124  001162            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4440  022502  001401                            BEQ     70$             ;BR IF OK FROM PORT A.
4441  022504  104007                            ERROR   7               ;REPORT ERROR
4442  022506  013737  001164  001126  70$:     MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
4443  022514  013737  001220  001226            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4444  022522  042737  100000  001164            BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
4445  022530  023737  001124  001164            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4446  022536  001401                            BEQ     71$             ;BR IF OK
4447  022540  104007                            ERROR   7               ;REPORT ERROR
4448  022542  000240                    71$:     NOP
4449
4450                                    ;;**********************************************************************
4451                                    ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT A)
4452
4453  022544  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
4454  022552  013737  001216  001226            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4455  022560  005037  001236                    CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
4456  022564  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4457  022572  012737  000012  001122            MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4458  022600  060037  001122                    ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
4459  022604  012737  100000  001124            MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4460  022612  013737  001126  001156            MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
4461  022620  042737  077777  001156            BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4462  022626  023737  001124  001156            CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
4463  022634  001414                            BEQ     72$              ;BR IF OK
4464  022636  013737  001126  001166            MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
4465  022644  042737  100000  001166            BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
4466  022652  053737  001166  001124            BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
4467  022660  104050                            ERROR   50               ;TYPE MESSAGE 50
4468  022662  005137  001236                    COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
4469  022666  000240                    72$:     NOP
4470  022670                            1$:
4471
4472                                    ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
4473
```

```
4474  022670  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
4475  022674  001412                          BEQ     TST16           ;;BR IF NOT
4476  022676  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
4477  022704  001406                          BEQ     TST16           ;;BR IF NOT
4478  022706  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
4479  022712  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
4480  022716  000177  156166                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
4481
4482
4483
4484                                  ;*****************************************************************
4485                                  ;*TEST 16       TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
4486                                  ;*
4487                                  ;*VERIFY THAT MASSBUS CLEAR DOES NOT RESET ATTENTION BITS WHEN THE
4488                                  ;*      DRIVE IS IN NEUTRAL.
4489                                  ;*
4490                                  ;*  A.  SET THE ATTENTION BITS FOR BOTH PORTS.
4491                                  ;*
4492                                  ;*  B.  VERIFY THAT THE DRIVE IS IN NEUTRAL.
4493                                  ;*
4494                                  ;*  C.  ISSUE A MASSBUS INIT.  VERIFY THAT NEITHER ATTENTION BIT HAS
4495                                  ;*      RESET.
4496                                  ;*
4497                                  ;*****************************************************************
4498  022722                          TST16:
4499  022722  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
4500  022724  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4501  022730  001406                          BEQ     2$              ;BR IF NOT
4502  022732  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
4503  022734  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
4504  022740  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4505  022746  112737  000016  001102  2$:     MOVB    #16,$TSTNM      ;TEST NUMBER
4506  022754  012737  022776  001106          MOV     #TEST16,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4507  022762  012737  022776  001110          MOV     #TEST16,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4508  022770  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
4509
4510
4511                                  ;;*************************************************************************
4512                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
4513
4514  022776                          TEST16:
4515                                  ;;*************************************************************************
4516
4517                                          ;SET ATTENTION BITS FOR BOTH PORTS
4518
4519  022776  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT 'A'
4520  023004  012760  177777  000014          MOV     #-1,RHER1(R0)   ;FORCE ERRORS
4521  023012  005060  000014                  CLR     RHER1(R0)       ;CLEAR THE ERRORS
4522  023016  013760  001220  000010          MOV     PORTB,RHCS2(R0) ;SELECT THE OTHER PORT
4523  023024  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
4524  023030  001775                          BEQ     .-4             ;BR IF DRIVE HASN'T TIMED OUT
4525  023032  012760  177777  000014          MOV     #-1,RHER1(R0)   ;FORCE ERRORS ON PORT 'B'
4526  023040  005060  000014                  CLR     RHER1(R0)       ;CLEAR THE ERRORS
4527  023044  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT "A" AGAIN
4528  023052  005760  000012                  TST     RHDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
4529  023056  001775                          BEQ     .-4             ;BR IF DRIVE HASN'T TIMED OUT
```

```
4530
4531                                    ;;*************************************************************************
4532                                    ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
4533
4534  023060  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
4535  023066  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4536  023074  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4537  023100  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4538  023106  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4539  023114  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4540  023120  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4541  023126  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4542  023134  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4543  023142  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4544  023150  001414                          BEQ     64$             ;BR IF OK
4545  023152  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4546  023160  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4547  023166  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4548  023174  104010                          ERROR   10              ;REPORT THE ERROR
4549  023176  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4550  023202  000240                  64$:    NOP
4551  023204  005737  001236                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
4552  023210  001402                          BEQ     .+6             ;BR IF IT WAS
4553  023212  000137  024156                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4554  023216  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4555  023224  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4556  023232  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4557  023236  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4558  023244  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4559  023252  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4560  023256  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4561  023264  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4562  023272  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4563  023300  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4564  023306  001414                          BEQ     65$             ;BR IF OK
4565  023310  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4566  023316  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4567  023324  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4568  023332  104010                          ERROR   10              ;REPORT THE ERROR
4569  023334  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4570  023340  000240                  65$:    NOP
4571  023342  005737  001236                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
4572  023346  001402                          BEQ     .+6             ;BR IF IT WAS
4573  023350  000137  024156                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4574
4575                                    ;;*************************************************************************
4576
4577                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4578
4579  023354  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4580  023360  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
4581  023366  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4582  023372  012737  111700  001124          MOV     #111700,$GDDAT  ;COMPARSION CONSTANT
4583  023400  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4584  023406  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4585  023414  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
4586  023422  042737  100100  001156          BIC     #ATA!VV,$TMPO   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4587  023430  113760  001220  000010          MOVB    PORTB,RHCS2(RO) ;SELECT PORT B.
4588  023436  016037  000012  001164          MOV     RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4589  023444  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4590  023452  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4591  023460  023737  001156  001160          CMP     $TMPO,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4592  023466  001006                          BNE     66$             ;BR IF NOT
4593  023470  005737  001156                  TST     $TMPO           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4594  023474  001045                          BNE     68$             ;BR IF NOT
4595  023476  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4596  023500  000137  023664                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
4597  023504  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4598  023512  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4599  023520  113760  001220  000010          MOVB    PORTB,RHCS2(RO) ;SELECT PORT B.
4600  023526  005737  001156                  TST     $TMPO           ;SEE IF STATUS EQ 0 FROM PORT A.
4601  023532  001414                          BEQ     67$             ;BR IF ZERO
4602  023534  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4603  023542  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4604  023550  113760  001216  000010          MOVB    PORTA,RHCS2(RO) ;SELECT PORT A.
4605  023556  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4606  023562  001012                          BNE     68$             ;BR IF NOT
4607  023564  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4608  023572  012760  000011  000000          MOV     #11,RHCS1(RO)   ;CLEAR THE DRIVE
4609  023600  012760  000013  000000          MOV     #13,RHCS1(RO)   ;RELEASE THE DRIVE
4610  023606  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4611  023610  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
4612  023616  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4613  023624  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4614  023632  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
4615  023634  104007                          ERROR   7               ;REPORT ERROR
4616  023636  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
4617  023644  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4618  023652  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4619  023660  001401                          BEQ     70$             ;BR IF OK
4620  023662  104007                          ERROR   7               ;REPORT ERROR
4621  023664  000240                  70$:    NOP
4622  023666  005737  001242                  TST     RELERR          ;WAS DRIVE IN NEUTRAL ?
4623  023672  001402                          BEQ     .+6             ;BR IF IT WAS
4624  023674  000137  024156                  JMP     1$              ;BYPASS RESET OF TEST
4625                                  ;;**************************************************************************
4626                                  ;ISSUE THE MASSBUS INIT
4627
4628  023700  012760  000040  000010          MOV     #CLR,RHCS2(RO)  ;ISSUE A MASSBUS INIT
4629
4630                                  ;;**************************************************************************
4631                                  ;CHECK THE ATTENTION BITS OF BOTH PORTS
4632
4633  023706  113760  001216  000010          MOVB    PORTA,RHCS2(RO) ;SELECT PORT A
4634  023714  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4635  023722  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4636  023726  016037  000012  001126          MOV     RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
4637  023734  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4638  023742  060037  001122                  ADD     RO,$BDADR       ;ADD RH11 BASE ADDRESS
4639  023746  012737  100000  001124          MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
4640  023754  013737  001126  001156          MOV     $BDDAT,$TMPO    ;MOVE REGISTER CONTENTS TO '$TMPO'
4641  023762  042737  077777  001156          BIC     #^CATA,$TMPO    ;SAVE SPECIFIED BITS
```

```
4642  023770  023737  001124  001156          CMP     $GDDAT,$TMPO    ;COMPARE THE BITS
4643  023776  001414                          BEQ     71$             ;BR IF OK
4644  024000  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4645  024006  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4646  024014  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4647  024022  104051                          ERROR   51              ;TYPE MESSAGE 51
4648  024024  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4649  024030  000240                  71$:    NOP
4650  024032  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4651  024040  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4652  024046  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4653  024052  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
4654  024060  012737  000012  001122          MO      #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4655  024066  060037  001122                  AD      R0,$BDADR       ;ADD RH11 BASE ADDRESS
4656  024072  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4657  024100  013737  001126  001156          MOV     $BDDAT,$TMPO    ;MOVE REGISTER CONTENTS TO '$TMPO'
4658  024106  042737  077777  001156          BIC     #↑CATA,$TMPO  ;SAVE SPECIFIED BITS
4659  024114  023737  001124  001156          CMP     $GDDAT,$TMPO    ;COMPARE THE BITS
4660  024122  001414                          BEQ     72$             ;BR IF OK
4661  024124  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4662  024132  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4663  024140  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4664  024146  104051                          ERROR   51              ;TYPE MESSAGE 51
4665  024150  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4666  024154  000240                  72$:    NOP
4667  024156                          1$:
4668
4669                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
4670
4671  024156  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
4672  024162  001412                          BEQ     TST17           ;;BR IF NOT
4673  024164  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
4674  024172  001406                          BEQ     TST17           ;;BR IF NOT
4675  024174  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
4676  024200  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
4677  024204  000177  154700                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
4678
4679
4680                                  ;*****************************************************************
4681                                  ;*TEST 17       TEST SEIZE BY RHCS1 READ THROUGH PORT 'A'
4682                                  ;*
4683                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RHCS1) SEIZES THE DRIVE.
4684                                  ;*
4685                                  ;*  A.  READ THE CONTROL REGISTER (RHCS1) THROUGH PORT 'A'; VERIFY THAT
4686                                  ;*      THE DRIVE IS SEIZED.
4687                                  ;*
4688                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE
4689                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4690                                  ;*
4691                                  ;*****************************************************************
4692  024210                          TST17:
4693  024210  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
4694  024212  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4695  024216  001406                          BEQ     2$              ;BR IF NOT
4696  024220  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
4697  024222  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
```

```
4698  024226  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4699  024234  112737  000017  001102  2$:     MOVB    #17,$TSTNM      ;TEST NUMBER
4700  024242  012737  024264  001106          MOV     #TEST17,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4701  024250  012737  024264  001110          MOV     #TEST17,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4702  024256  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
4703
4704
4705                                  ;;*****************************************************************************
4706                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
4707
4708  024264                          TEST17:
4709
4710                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
4711
4712  024264  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
4713  024272  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
4714  024276  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
4715  024304  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4716  024312  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
4717  024320  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
4718  024324  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
4719  024332  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4720
4721                                  ;;*****************************************************************************
4722
4723                                          ;SEIZE THE DRIVE THROUGH PORT A
4724
4725  024340  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
4726  024346  013737  001216  001230          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
4727  024354  005760  000000                  TST     RHCS1(R0)       ;READ RHCS1
4728  024360  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
4729  024366  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4730  024374  013737  001220  001232          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
4731  024402  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
4732  024410  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
4733  024414  062737  000012  001122          ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
4734  024422  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
4735  024426  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
4736  024434  001403                          BEQ     .+10            ;BR IF IT IS
4737  024436  104004                          ERROR   4               ;REPORT THE ERROR
4738  024440  000137  025070                  JMP     1$              ;BYPASS REST OF THE SUBTEST
4739  024444  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
4740  024452  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4741  024460  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
4742  024466  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
4743  024474  013737  001124  001160          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
4744  024502  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
4745  024506  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
4746  024514  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
4747  024522  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
4748  024530  001401                          BEQ     .+4             ;BR IF THEY ARE
4749  024532  104005                          ERROR 5                 ;REPORT THE ERROR
4750
4751                                  ;;*****************************************************************************
4752
4753                                          ;RELEASE THE DRIVE FROM PORT A
```

```
4754
4755  024534  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
4756  024542  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4757  024550  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
4758
4759                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4760
4761  024556  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4762  024562  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
4763  024570  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4764  024574  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
4765  024602  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4766  024610  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4767  024616  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
4768  024624  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4769  024632  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4770  024640  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4771  024646  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4772  024654  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4773  024662  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4774  024670  001006                          BNE     64$             ;BR IF NOT
4775  024672  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4776  024676  001045                          BNE     66$             ;BR IF NOT
4777  024700  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4778  024702  000137  025066                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
4779  024706  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4780  024714  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4781  024722  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4782  024730  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4783  024734  001414                          BEQ     65$             ;BR IF ZERO
4784  024736  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4785  024744  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4786  024752  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4787  024760  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4788  024764  001012                          BNE     66$             ;BR IF NOT
4789  024766  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4790  024774  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
4791  025002  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4792  025010  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4793  025012  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
4794  025020  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4795  025026  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4796  025034  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
4797  025036  104007                          ERROR   7               ;REPORT ERROR
4798  025040  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
4799  025046  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4800  025054  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4801  025062  001401                          BEQ     68$             ;BR IF OK
4802  025064  104007                          ERROR   7               ;REPORT ERROR
4803  025066  000240                  68$:    NOP
4804  025070                          1$:
4805
4806                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
4807
4808  025070  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
4809  025074  001412                          BEQ     TST20           ;;BR IF NOT
```

```
 4810  025076  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
 4811  025104  001406                          BEQ     TST20           ;;BR IF NOT
 4812  025106  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
 4813  025112  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
 4814  025116  000177  153766                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
 4815
 4816                                  ;*****************************************************************
 4817                                  ;*TEST 20       TEST SEIZE BY RHCS1 READ THROUGH PORT 'B'
 4818                                  ;*
 4819                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RHCS1) SEIZES THE DRIVE.
 4820                                  ;*
 4821                                  ;*  A.  READ THE CONTROL REGISTER (RHCS1) THROUGH PORT 'B'; VERIFY THAT
 4822                                  ;*      THE DRIVE IS SEIZED.
 4823                                  ;*
 4824                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
 4825                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 4826                                  ;*
 4827                                  ;*****************************************************************
 4828  025122                          TST20:
 4829  025122  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
 4830  025124  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4831  025130  001406                          BEQ     2$              ;BR IF NOT
 4832  025132  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 4833  025134  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4834  025140  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4835  025146  112737  000020  001102  2$:     MOVB    #20,$TSTNM      ;TEST NUMBER
 4836  025154  012737  025176  001106          MOV     #TEST20,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4837  025162  012737  025176  001110          MOV     #TEST20,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4838  025170  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 4839
 4840
 4841
 4842                                  ;;*************************************************************************
 4843                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
 4844
 4845  025176                          TEST20:
 4846
 4847                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 4848
 4849  025176  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
 4850  025204  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
 4851  025210  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
 4852  025216  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 4853  025224  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
 4854  025232  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 4855  025236  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
 4856  025244  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 4857
 4858                                  ;;*************************************************************************
 4859
 4860                                          ;SEIZE THE DRIVE THROUGH PORT B
 4861
 4862  025252  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
 4863  025260  013737  001220  001230          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 4864  025266  005760  000000                  TST     RHCS1(R0)       ;READ RHCS1
 4865  025272  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
```

```
4866  025300  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4867  025306  013737  001216  001232          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
4868  025314  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;SEE IF DRIVE SEIZED BY PORT B
4869  025322  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
4870  025326  062737  000012  001122          ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
4871  025334  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
4872  025340  023737  001124  001125          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
4873  025346  001403                          BEQ     .+10            ;BR IF IT IS
4874  025350  104004                          ERROR   4               ;REPORT THE ERROR
4875  025352  000137  026002                  JMP     1$              ;BYPASS REST OF THE SUBTEST
4876  025356  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
4877  025364  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4878  025372  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
4879  025400  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
4880  025406  013737  001124  001160          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
4881  025414  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
4882  025420  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
4883  025426  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
4884  025434  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
4885  025442  001401                          BEQ     .+4             ;BR IF THEY ARE
4886  025444  104005                          ERROR 5                 ;REPORT THE ERROR
4887
4888                                  ;;**************************************************************************
4889
4890                                          ;RELEASE THE DRIVE FROM PORT B
4891
4892  025446  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
4893  025454  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4894  025462  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
4895
4896                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4897
4898  025470  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4899  025474  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
4900  025502  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4901  025506  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
4902  025514  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
4903  025522  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4904  025530  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
4905  025536  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4906  025544  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4907  025552  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4908  025560  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4909  025566  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4910  025574  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4911  025602  001006                          BNE     64$             ;BR IF NOT
4912  025604  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4913  025610  001045                          BNE     66$             ;BR IF NOT
4914  025612  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4915  025614  000137  026000                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
4916  025620  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4917  025626  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4918  025634  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
4919  025642  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4920  025646  001414                          BEQ     65$             ;BR IF ZERO
4921  025650  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
 4922  025656  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 4923  025664  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
 4924  025672  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 4925  025676  001012                          BNE     66$             ;BR IF NOT
 4926  025700  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 4927  025706  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
 4928  025714  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 4929  025722  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 4930  025724  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
 4931  025732  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 4932  025740  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 4933  025746  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
 4934  025750  104007                          ERROR   7               ;REPORT ERROR
 4935  025752  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
 4936  025760  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 4937  025766  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 4938  025774  001401                          BEQ     68$             ;BR IF OK
 4939  025776  104007                          ERROR   7               ;REPORT ERROR
 4940  026000  000240                  68$:    NOP
 4941  026002                          1$:
 4942
 4943                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
 4944
 4945  026002  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
 4946  026006  001412                          BEQ     TST21           ;;BR IF NOT
 4947  026010  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
 4948  026016  001406                          BEQ     TST21           ;;BR IF NOT
 4949  026020  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
 4950  026024  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
 4951  026030  000177  153054                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
 4952
 4953
 4954                          ;*****************************************************************
 4955                          ;*TEST 21       TEST 'PORT REQUEST' FROM PORT 'A'
 4956                          ;*
 4957                          ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
 4958                          ;*      DRIVE IS SEIZED BY THE OTHER PORT.
 4959                          ;*
 4960                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
 4961                          ;*
 4962                          ;*  B.  WRITE 0'S INTO RHDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
 4963                          ;*      SEIZED BY PORT 'B'.
 4964                          ;*
 4965                          ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
 4966                          ;*      SWITCHED TO PORT 'A'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
 4967                          ;*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
 4968                          ;*
 4969                          ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
 4970                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 4971                          ;*
 4972                          ;*****************************************************************
 4973  026034                  TST21:
 4974  026034  000004                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
 4975  026036  005737  001266          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4976  026042  001406                  BEQ     2$              ;BR IF NOT
 4977  026044  100002                  BPL     1$              ;BR IF JUST ENTERED TEST
```

```
4978  026046  000137  002410                    JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
4979  026052  012737  177777  001266  1$:       MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
4980  026060  112737  000021  001102  2$:       MOVB    #21,$TSTNM              ;TEST NUMBER
4981  026066  012737  026110  001106            MOV     #TEST21,$LPADR          ;LOAD LOOP ON TEST ADDRESS
4982  026074  012737  026110  001110            MOV     #TEST21,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
4983  026102  012737  007640  001170            MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS
4984
4985
4986                                    ;;**************************************************************************
4987                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
4988
4989  026110                            TEST21:
4990
4991                                            ;CLEAR ATTENTION BITS FOR BOTH PORTS
4992
4993  026110  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
4994  026116  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE
4995  026122  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
4996  026130  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
4997  026136  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
4998  026144  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
4999  026150  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5000  026156  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5001
5002                                    ;;**************************************************************************
5003
5004                                            ;SEIZE THE DRIVE THROUGH PORT B
5005
5006  026164  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5007  026172  013737  001220  001230            MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
5008  026200  005060  000012                    CLR     RHDS1(R0)       ;WRITE RHDS1
5009  026204  013737  001216  001232            MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
5010  026212  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
5011  026220  013737  001216  001226            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5012
5013                                    ;;**************************************************************************
5014                                    ;SET PORT REQUEST
5015
5016  026226  005060  000012                    CLR     RHDS1(R0)       ;SET PORT REQUEST FOR PORT A
5017
5018                                    ;;**************************************************************************
5019                                    ;RELEASE THROUGH PORT B.  DRIVE SHOULD SWITCH TO PORT A.
5020
5021
5022                                            ;RELEASE THE DRIVE FROM PORT B
5023
5024  026232  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5025  026240  013737  001220  001226            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5026  026246  012760  000013  000000            MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5027
5028                                            ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5029
5030  026254  005037  001242                    CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5031  026260  012737  111700  001124            MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5032  026266  012737  000012  001122            MOV     #RHDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
5033  026274  060037  001122                    ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
```

```
5034  026300  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5035  026306  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5036  026314  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
5037  026322  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5038  026330  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5039  026336  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
5040  026344  001404                          BEQ     64$             ;BR IF STATUS FROM PORT B ZERO
5041  026346  005737  001156                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
5042  026352  001401                          BEQ     64$             ;BR IF ZERO
5043  026354  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
5044  026356  013737  001156  001126  64$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
5045  026364  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5046  026372  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5047  026400  001401                          BEQ     .+4             ;BR IF OK
5048  026402  104027                          ERROR   27              ;REPORT REGISTER ERROR
5049  026404  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5050  026412  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5051  026420  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5052  026424  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
5053  026432  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5054  026440  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5055  026444  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5056  026450  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5057  026456  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5058  026464  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5059  026472  001414                          BEQ     65$             ;BR IF OK
5060  026474  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5061  026502  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5062  026510  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5063  026516  104016                          ERROR   16              ;TYPE MESSAGE 16
5064  026520  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5065  026524  000240                  65$:    NOP
5066  026526  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5067  026534  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5068  026542  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5069  026546  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
5070  026554  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5071  026562  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5072  026566  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
5073  026574  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5074  026602  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5075  026610  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5076  026616  001414                          BEQ     66$             ;BR IF OK
5077  026620  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5078  026626  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5079  026634  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5080  026642  104016                          ERROR   16              ;TYPE MESSAGE 16
5081  026644  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5082  026650  000240                  66$:    NOP
5083
5084                                  ;;*******************************************************************************
5085
5086                                          ;RELEASE THE DRIVE FROM PORT A
5087
5088  026652  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5089  026660  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
5090  026666  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5091
5092                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5093
5094  026674  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5095  026700  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
5096  026706  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5097  026712  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5098  026720  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5099  026726  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5100  026734  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5101  026742  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5102  026750  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5103  026756  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5104  026764  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5105  026772  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5106  027000  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5107  027006  001006                          BNE     67$             ;BR IF NOT
5108  027010  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5109  027014  001045                          BNE     69$             ;BR IF NOT
5110  027016  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5111  027020  000137  027204                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
5112  027024  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5113  027032  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5114  027040  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5115  027046  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5116  027052  001414                          BEQ     68$             ;BR IF ZERO
5117  027054  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5118  027062  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5119  027070  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5120  027076  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5121  027102  001012                          BNE     69$             ;BR IF NOT
5122  027104  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5123  027112  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
5124  027120  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5125  027126  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5126  027130  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
5127  027136  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5128  027144  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5129  027152  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
5130  027154  104007                          ERROR   7               ;REPORT ERROR
5131  027156  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
5132  027164  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5133  027172  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5134  027200  001401                          BEQ     71$             ;BR IF OK
5135  027202  104007                          ERROR   7               ;REPORT ERROR
5136  027204  000240                  71$:    NOP
5137  027206                          1$:
5138
5139                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
5140
5141  027206  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
5142  027212  001412                          BEQ     TST22           ;;BR IF NOT
5143  027214  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
5144  027222  001406                          BEQ     TST22           ;;BR IF NOT
5145  027224  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
```

```
5146  027230  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
5147  027234  000177  151650                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
5148
5149                                  ;*****************************************************************
5150                                  ;*TEST 22       TEST PORT REQUEST FROM PORT 'B'
5151                                  ;*
5152                                  ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
5153                                  ;*      DRIVE IS SEIZED BY THE OTHER PORT.
5154                                  ;*
5155                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
5156                                  ;*
5157                                  ;*  B.  WRITE 0'S INTO RHDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
5158                                  ;*      SEIZED BY PORT 'A'.
5159                                  ;*
5160                                  ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
5161                                  ;*      SWITCHED TO PORT 'B'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
5162                                  ;*      PORT 'B' AND IS NOT SET FOR PORT 'A'.
5163                                  ;*
5164                                  ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
5165                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5166                                  ;*
5167                                  ;*****************************************************************
5168  027240                          TST22:
5169  027240  000004                         SCOPE                   ;INITIALIZE THE SCOPE HANDLER
5170  027242  005737  001266                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
5171  027246  001406                         BEQ     2$              ;BR IF NOT
5172  027250  100002                         BPL     1$              ;BR IF JUST ENTERED TEST
5173  027252  000137  002410                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
5174  027256  012737  177777  001266  1$:    MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
5175  027264  112737  000022  001102  2$:    MOVB    #22,$TSTNM      ;TEST NUMBER
5176  027272  012737  027314  001106         MOV     #TEST22,$LPADR  ;LOAD LOOP ON TEST ADDRESS
5177  027300  012737  027314  001110         MOV     #TEST22,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
5178  027306  012737  007640  001170         MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
5179
5180
5181                                  ;;*************************************************************************
5182                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
5183
5184  027314                          TEST22:
5185
5186                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
5187
5188  027314  113760  001216  000010         MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
5189  027322  005060  000012                 CLR     RHDS1(R0)       ;SEIZE THE DRIVE
5190  027326  012760  000011  000000         MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5191  027334  012760  000013  000000         MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5192  027342  113760  001220  000010         MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
5193  027350  005060  000012                 CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
5194  027354  012760  000011  000000         MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5195  027362  012760  000013  000000         MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5196
5197                                  ;;*************************************************************************
5198
5199                                          ;SEIZE THE DRIVE THROUGH PORT A
5200
5201  027370  113760  001216  000010         MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
```

```
5202  027376  013737  001216  001230          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
5203  027404  005060  000012                  CLR     RHDS1(R0)     ;WRITE RHDS1
5204  027410  013737  001220  001232          MOV     PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
5205  027416  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5206  027424  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5207
5208                                  ;;**********************************************************************
5209                                  ;SET PORT REQUEST
5210
5211  027432  005060  000012                  CLR     RHDS1(R0)     ;SET PORT REQUEST FOR PORT B
5212
5213                                  ;;**********************************************************************
5214                                  ;RELEASE THROUGH PORT A.  DRIVE SHOULD SWITCH TO PORT B.
5215
5216
5217                                          ;RELEASE THE DRIVE FROM PORT A
5218
5219  027436  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5220  027444  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5221  027452  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5222
5223                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5224
5225  027460  005037  001242                  CLR     RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
5226  027464  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5227  027472  012737  000012  001122          MOV     #RHDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5228  027500  060037  001122                  ADD     R0,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
5229  027504  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5230  027512  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5231  027520  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
5232  027526  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5233  027534  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5234  027542  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
5235  027550  001404                          BEQ     64$           ;BR IF STATUS FROM PORT A ZERO
5236  027552  005737  001156                  TST     $TMP0         ;IS STATUS FROM PORT B ZERO ?
5237  027556  001401                          BEQ     64$           ;BR IF ZERO
5238  027560  104031                          ERROR   31            ;REPORT DRIVE IN NEUTRAL
5239  027562  013737  001156  001126  64$:    MOV     $TMP0,$BDDAT  ;CHECK STATUS FROM PORT B
5240  027570  013737  001220  001226          MOV     PORTB,PTNBR   ;CHANGE PORT ADDRESS FOR TYPEOUT
5241  027576  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
5242  027604  001401                          BEQ     .+4           ;BR IF OK
5243  027606  104027                          ERROR   27            ;REPORT REGISTER ERROR
5244  027610  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5245  027616  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5246  027624  005037  001236                  CLR     CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
5247  027630  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
5248  027636  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5249  027644  060037  001122                  ADD     R0,$BDADR     ;ADD RH11 BASE ADDRESS
5250  027650  005037  001124                  CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
5251  027654  013737  001126  001156          MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
5252  027662  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5253  027670  023737  001124  001156          CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
5254  027676  001414                          BEQ     65$           ;BR IF OK
5255  027700  013737  001126  001166          MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
5256  027706  042737  100000  001166          BIC     #ATA,$TMP4    ;CLEAR THE MASKED BITS
5257  027714  053737  001166  001124          BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
5258  027722  104016                          ERROR   16              ;TYPE MESSAGE 16
5259  027724  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5260  027730  000240                  65$:    NOP
5261  027732  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5262  027740  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5263  027746  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5264  027752  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
5265  027760  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5266  027766  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5267  027772  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
5268  030000  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5269  030006  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5270  030014  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5271  030022  001414                          BEQ     66$             ;BR IF OK
5272  030024  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5273  030032  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5274  030040  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5275  030046  104016                          ERROR   16              ;TYPE MESSAGE 16
5276  030050  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5277  030054  000240                  66$:    NOP
5278
5279                                  ;;*****************************************************************************
5280
5281                                          ;RELEASE THE DRIVE FROM PORT B
5282
5283  030056  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5284  030064  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5285  030072  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5286
5287                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5288
5289  030100  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5290  030104  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
5291  030112  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5292  030116  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5293  030124  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5294  030132  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5295  030140  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5296  030146  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5297  030154  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5298  030162  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5299  030170  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5300  030176  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5301  030204  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5302  030212  001006                          BNE     67$             ;BR IF NOT
5303  030214  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5304  030220  001045                          BNE     69$             ;BR IF NOT
5305  030222  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5306  030224  000137  030410                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
5307  030230  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5308  030236  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5309  030244  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5310  030252  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5311  030256  001414                          BEQ     68$             ;BR IF ZERO
5312  030260  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5313  030266  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
```

```
5314  030274  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5315  030302  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5316  030306  001012                          BNE     69$             ;BR IF NOT
5317  030310  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5318  030316  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
5319  030324  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5320  030332  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5321  030334  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
5322  030342  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5323  030350  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5324  030356  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
5325  030360  104007                          ERROR   7               ;REPORT ERROR
5326  030362  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
5327  030370  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5328  030376  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5329  030404  001401                          BEQ     71$             ;BR IF OK
5330  030406  104007                          ERROR   7               ;REPORT ERROR
5331  030410  000240                  71$:    NOP
5332  030412                          1$:
5333
5334                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
5335
5336  030412  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
5337  030416  001412                          BEQ     TST23           ;;BR IF NOT
5338  030420  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
5339  030426  001406                          BEQ     TST23           ;;BR IF NOT
5340  030430  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
5341  030434  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
5342  030440  000177  150444                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
5343
5344
5345                                  ;*****************************************************************
5346                                  ;*TEST 23       TEST NO 'PORT REQUEST' WHEN READ RHCS1 THROUGH PORT 'A'
5347                                  ;*
5348                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RHCS1) DOES NOT SET 'PORT
5349                                  ;*      REQUEST'.
5350                                  ;*
5351                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RHCS1.  VERIFY THAT
5352                                  ;*      THE DRIVE HAS BEEN SEIZED.
5353                                  ;*
5354                                  ;*  B.  READ THE CONTROL REGISTER FROM PORT 'A'.  VERIFY THAT 'DVA' IS NOT
5355                                  ;*      SET.
5356                                  ;*
5357                                  ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
5358                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5359                                  ;*
5360                                  ;*****************************************************************
5361  030444                          TST23:
5362  030444  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
5363  030446  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
5364  030452  001406                          BEQ     2$              ;BR IF NOT
5365  030454  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
5366  030456  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
5367  030462  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
5368  030470  112737  000023  001102  2$:     MOVB    #23,$TSTNM      ;TEST NUMBER
5369  030476  012737  030520  001106          MOV     #TEST23,$LPADR  ;LOAD LOOP ON TEST ADDRESS
```

```
5370  030504  012737  030520  001110          MOV     #TEST23,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
5371  030512  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
5372
5373
5374                                  ;;**********************************************************************
5375                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
5376
5377  030520                          TEST23:
5378
5379                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
5380
5381  030520  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
5382  030526  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
5383  030532  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5384  030540  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5385  030546  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
5386  030554  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
5387  030560  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5388  030566  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5389
5390                                  ;;**********************************************************************
5391
5392                                          ;SEIZE THE DRIVE THROUGH PORT B
5393
5394  030574  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5395  030602  013737  001220  001230          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
5396  030610  005760  000000                  TST     RHCS1(R0)       ;READ RHCS1
5397  030614  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
5398  030622  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5399  030630  013737  001216  001232          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
5400  030636  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
5401  030644  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
5402  030650  062737  000012  001122          ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
5403  030656  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
5404  030662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
5405  030670  001403                          BEQ     .+10            ;BR IF IT IS
5406  030672  104004                          ERROR   4               ;REPORT THE ERROR
5407  030674  000137  031446                  JMP     1$              ;BYPASS REST OF THE SUBTEST
5408  030700  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5409  030706  013737  001220  001226          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5410  030714  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5411  030722  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5412  030730  013737  001124  001160          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
5413  030736  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
5414  030742  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
5415  030750  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
5416  030756  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
5417  030764  001401                          BEQ     .+4             ;BR IF THEY ARE
5418  030766  104005                          ERROR 5                 ;REPORT THE ERROR
5419  030770  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
5420  030776  013737  001216  001226          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5421
5422                                  ;;**********************************************************************
5423                                  ;READ RHCS1 THROUGH PORT A - TRY TO SET PORT REQUEST
5424
5425  031004  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
5426  031010  016037  000000  001126          MOV     RHCS1(R0),$BDDAT  ;GET CONTENTS OF RHCS1
5427  031016  012737  000000  001122          MOV     #RHCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5428  031024  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5429  031030  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5430  031034  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5431  031042  042737  173700  001156          BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
5432  031050  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5433  031056  001414                          BEQ     64$             ;BR IF OK
5434  031060  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5435  031066  042737  004077  001166          BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
5436  031074  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5437  031102  104010                          ERROR   10              ;REPORT THE ERROR
5438  031104  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5439  031110  000240                  64$:    NOP
5440
5441                                  ;;*************************************************************************
5442                                  ;DRIVE SHOULD RETURN TO NEUTRAL
5443
5444
5445                                          ;RELEASE THE DRIVE FROM PORT B
5446
5447  031112  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5448  031120  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5449  031126  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5450
5451                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5452
5453  031134  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5454  031140  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
5455  031146  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5456  031152  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5457  031160  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5458  031166  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5459  031174  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5460  031202  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5461  031210  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5462  031216  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5463  031224  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5464  031232  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5465  031240  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5466  031246  001006                          BNE     65$             ;BR IF NOT
5467  031250  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5468  031254  001045                          BNE     67$             ;BR IF NOT
5469  031256  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5470  031260  000137  031444                  JMP     69$             ;BYPASS THE REST OF THE CHECKS
5471  031264  013737  001162  001126  65$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5472  031272  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5473  031300  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5474  031306  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5475  031312  001414                          BEQ     66$             ;BR IF ZERO
5476  031314  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5477  031322  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5478  031330  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5479  031336  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5480  031342  001012                          BNE     67$             ;BR IF NOT
5481  031344  012737  177777  001242  66$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
```

```
5482  031352  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
5483  031360  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5484  031366  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5485  031370  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
5486  031376  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5487  031404  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5488  031412  001401                          BEQ     68$             ;BR IF OK FROM PORT A.
5489  031414  104007                          ERROR   7               ;REPORT ERROR
5490  031416  013737  001164  001126  68$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
5491  031424  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5492  031432  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5493  031440  001401                          BEQ     69$             ;BR IF OK
5494  031442  104007                          ERROR   7               ;REPORT ERROR
5495  031444  000240                  69$:    NOP
5496  031446                          1$:
5497
5498                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
5499
5500  031446  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
5501  031452  001412                          BEQ     TST24           ;;BR IF NOT
5502  031454  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
5503  031462  001406                          BEQ     TST24           ;;BR IF NOT
5504  031464  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
5505  031470  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
5506  031474  000177  147410                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
5507
5508                                  ;*****************************************************************
5509                                  ;*TEST 24       TEST NO 'PORT REQUEST' WHEN READ RHCS1 THROUGH PORT 'B'
5510                                  ;*
5511                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RHCS1) DOES NOT SET 'PORT
5512                                  ;*      REQUEST'.
5513                                  ;*
5514                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY READING RHCS1.  VERIFY THAT
5515                                  ;*      THE DRIVE HAS BEEN SEIZED.
5516                                  ;*
5517                                  ;*  B.  READ THE CONTROL REGISTER FROM PORT 'B'.  VERIFY THAT 'DVA' IS NOT
5518                                  ;*      SET.
5519                                  ;*
5520                                  ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
5521                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5522                                  ;*
5523                                  ;*****************************************************************
5524  031500                          TST24:
5525  031500  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
5526  031502  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
5527  031506  001406                          BEQ     2$              ;BR IF NOT
5528  031510  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
5529  031512  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
5530  031516  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
5531  031524  112737  000024  001102  2$:     MOVB    #24,$TSTNM      ;TEST NUMBER
5532  031532  012737  031554  001106          MOV     #TEST24,$LPADR  ;LOAD LOOP ON TEST ADDRESS
5533  031540  012737  031554  001110          MOV     #TEST24,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
5534  031546  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
5535
5536
5537                                  ;;*****************************************************************************
```

```
                                        ;END OF 'SCOPE' SETUP - START OF MAIN TEST
                                        TEST24:
                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS
031554  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
031562  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
031566  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
031574  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
031602  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
031610  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
031614  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
031622  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;;**************************************************************************
                                                ;SEIZE THE DRIVE THROUGH PORT A
031630  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
031636  013737  001216  001230                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
031644  005760  000000                          TST     RHCS1(R0)       ;READ RHCS1
031650  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
031656  013737  001220  001226                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031664  013737  001220  001232                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
031672  016037  000012  001126                  MOV     RHDS1'R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
031700  010037  001122                          MOV     R0,$BDADR       ;RH11 BASE ADDRESS
031704  062737  000012  001122                  ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
031712  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
031716  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
031724  001403                                  BEQ     .+10            ;BR IF IT IS
031726  104004                                  ERROR   4               ;REPORT THE ERROR
031730  000137  032502                          JMP     1$              ;BYPASS REST OF THE SUBTEST
031734  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
031742  013737  001216  001226                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031750  016037  000012  001126                  MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
031756  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
031764  013737  001124  001160                  MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
031772  005137  001160                          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
031776  013737  001126  001156                  MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
032004  043737  001160  001156                  BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
032012  023737  001124  001156                  CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
032020  001401                                  BEQ     .+4             ;BR IF THEY ARE
032022  104005                                  ERROR 5                 ;REPORT THE ERROR
032024  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
032032  013737  001220  001226                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;**************************************************************************
                                        ;READ RHCS1 THROUGH PORT B - TRY TO SET PORT REQUEST

032040  005037  001236                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
032044  016037  000000  001126                  MOV     RHCS1(R0),$BDDAT ;GET CONTENTS OF RHCS1
032052  012737  000000  001122                  MOV     #RHCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032060  060037  001122                          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
032064  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
032070  013737  001126  001156                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
```

```
5594  032076  042737  173700  001156          BIC     #^C4077,$TMP0   ;SAVE SPECIFIED BITS
5595  032104  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5596  032112  001414                          BEQ     64$             ;BR IF OK
5597  032114  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5598  032122  042737  004077  001166          BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
5599  032130  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5600  032136  104010                          ERROR   10              ;REPORT THE ERROR
5601  032140  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5602  032144  000240                  64$:    NOP
5603
5604                                  ;;*************************************************************************
5605                                  ;DRIVE SHOULD RETURN TO NEUTRAL
5606
5607
5608                                          ;RELEASE THE DRIVE FROM PORT A
5609
5610  032146  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5611  032154  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5612  032162  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5613
5614                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5615
5616  032170  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5617  032174  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
5618  032202  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5619  032206  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5620  032214  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5621  032222  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5622  032230  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5623  032236  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5624  032244  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5625  032252  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5626  032260  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5627  032266  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5628  032274  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5629  032302  001006                          BNE     65$             ;BR IF NOT
5630  032304  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5631  032310  001045                          BNE     67$             ;BR IF NOT
5632  032312  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5633  032314  000137  032500                  JMP     69$             ;BYPASS THE REST OF THE CHECKS
5634  032320  013737  001162  001126  65$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5635  032326  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5636  032334  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5637  032342  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5638  032346  001414                          BEQ     66$             ;BR IF ZERO
5639  032350  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5640  032356  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5641  032364  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5642  032372  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5643  032376  001012                          BNE     67$             ;BR IF NOT
5644  032400  012737  177777  001242  66$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5645  032406  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
5646  032414  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5647  032422  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5648  032424  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
5649  032432  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
```

```
5650  032440  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5651  032446  001401                          BEQ     68$             ;BR IF OK FROM PORT A.
5652  032450  104007                          ERROR   7               ;REPORT ERROR
5653  032452  013737  001164  001126  68$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
5654  032460  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5655  032466  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5656  032474  001401                          BEQ     69$             ;BR IF OK
5657  032476  104007                          ERROR   7               ;REPORT ERROR
5658  032500  000240                  69$:    NOP
5659  032502                          1$:
5660
5661                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
5662
5663  032502  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
5664  032506  001412                          BEQ     TST25           ;;BR IF NOT
5665  032510  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
5666  032516  001406                          BEQ     TST25           ;;BR IF NOT
5667  032520  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
5668  032524  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
5669  032530  000177  146354                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
5670
5671
5672                                  ;*******************************************************************
5673                                  ;*TEST 25       TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
5674                                  ;*
5675                                  ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
5676                                  ;*      IS SEIZED BY THE OTHER PORT.
5677                                  ;*
5678                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
5679                                  ;*
5680                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
5681                                  ;*
5682                                  ;*  C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
5683                                  ;*
5684                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE SWITCHED
5685                                  ;*      TO PORT 'A'.
5686                                  ;*
5687                                  ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
5688                                  ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5689                                  ;*
5690                                  ;*******************************************************************
5691  032534                          TST25:
5692  032534  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
5693  032536  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
5694  032542  001406                          BEQ     2$              ;BR IF NOT
5695  032544  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
5696  032546  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
5697  032552  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
5698  032560  112737  000025  001102  2$:     MOVB    #25,$TSTNM      ;TEST NUMBER
5699  032566  012737  032610  001106          MOV     #TST25,$LPADR   ;LOAD LOOP ON TEST ADDRESS
5700  032574  012737  032610  001110          MOV     #TST25,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
5701  032602  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
5702
5703
5704                                  ;;*******************************************************************************
5705                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
```

```
5706
5707  032610                                TEST25:
5708
5709                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS
5710
5711  032610  113760  001216  000010                MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
5712  032616  005060  000012                        CLR     RHDS1(R0)       ;SEIZE THE DRIVE
5713  032622  012760  000011  000000                MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5714  032630  012760  000013  000000                MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5715  032636  113760  001220  000010                MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
5716  032644  005060  000012                        CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
5717  032650  012760  000011  000000                MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
5718  032656  012760  000013  000000                MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5719
5720                                        ;;**************************************************************************
5721
5722                                                ;SEIZE THE DRIVE THROUGH PORT B
5723
5724  032664  113760  001220  000010                MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5725  032672  013737  001220  001230                MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
5726  032700  005060  000012                        CLR     RHDS1(R0)       ;WRITE RHDS1
5727  032704  113760  001216  000010                MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
5728  032712  013737  001216  001226                MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5729  032720  013737  001216  001232                MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
5730  032726  016037  000012  001126                MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
5731  032734  010037  001122                        MOV     R0,$BDADR       ;RH11 BASE ADDRESS
5732  032740  062737  000012  001122                ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
5733  032746  005037  001124                        CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
5734  032752  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
5735  032760  001403                                BEQ     .+10            ;BR IF IT IS
5736  032762  104004                                ERROR   4               ;REPORT THE ERROR
5737  032764  000137  033716                        JMP     1$              ;BYPASS REST OF THE SUBTEST
5738  032770  113760  001220  000010                MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
5739  032776  013737  001220  001226                MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5740  033004  016037  000012  001126                MOV     RHDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
5741  033012  012737  011700  001124                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
5742  033020  013737  001124  001160                MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
5743  033026  005137  001160                        COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
5744  033032  013737  001126  001156                MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
5745  033040  043737  001160  001156                BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
5746  033046  023737  001124  001156                CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
5747  033054  001401                                BEQ     .+4             ;BR IF THEY ARE
5748  033056  104005                                ERROR 5                 ;REPORT THE ERROR
5749
5750                                        ;;**************************************************************************
5751                                        ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
5752
5753  033060  113760  001216  000010                MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
5754  033066  013737  001216  001226                MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5755  033074  012760  000013  000000                MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT A
5756
5757                                        ;;**************************************************************************
5758                                        ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
5759
5760  033102  005037  001236                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5761  033106  016037  000012  001126                MOV     RHDS1(R0),$BDDAT ;GET CONTENTS OF RHDS1
```

```
5762  033114  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5763  033122  060037  001122                  ADD     R0,$BDADR      ;ADD RH11 BASE ADDRESS
5764  033126  005037  001124                  CLR     $GDDAT         ;WHAT REGISTER SHOULD BE
5765  033132  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;IS THE REGISTER OK ?
5766  033140  001403                          BEQ     64$            ;BR IF OK
5767  033142  104010                          ERROR   10             ;REPORT THE ERROR
5768  033144  005137  001236                  COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
5769  033150  016037  000000  001126  64$:    MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
5770  033156  032737  020000  001126          BIT     #MCPE,$BDDAT   ;IS 'MCPE' SET ?
5771  033164  001404                          BEQ     .+12           ;BR IF NOT
5772  033166  104011                          ERROR   11             ;REPORT THE ERROR
5773  033170  012760  040000  000000          MOV     #TRE,RHCS1(R0) ;CLEAR 'MCPE'
5774  033176  005737  001236                  TST     CKERR          ;WAS RHDS1 NON ZERO ?
5775  033202  001402                          BEQ     .+6            ;CONTENTS OF RHDS1 SEEN BY PORT A
5776  033204  000137  033716                  JMP     1$             ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5777
5778                                  ;;**********************************************************************
5779
5780                                  ;RELEASE THE DRIVE FROM PORT B
5781
5782  033210  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5783  033216  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5784  033224  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5785
5786                                  ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5787
5788  033232  005037  001242                  CLR     RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
5789  033236  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5790  033244  012737  000012  001122          MOV     #RHDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5791  033252  060037  001122                  ADD     R0,$BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
5792  033256  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5793  033264  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5794  033272  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
5795  033300  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5796  033306  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5797  033314  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
5798  033322  001404                          BEQ     65$            ;BR IF STATUS FROM PORT B ZERO
5799  033324  005737  001156                  TST     $TMP0          ;IS STATUS FROM PORT A ZERO ?
5800  033330  001401                          BEQ     65$            ;BR IF ZERO
5801  033332  104031                          ERROR   31             ;REPORT DRIVE IN NEUTRAL
5802  033334  013737  001156  001126  65$:    MOV     $TMP0,$BDDAT   ;CHECK STATUS FROM PORT A
5803  033342  013737  001216  001226          MOV     PORTA,PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
5804  033350  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
5805  033356  001401                          BEQ     .+4            ;BR IF OK
5806  033360  104027                          ERROR   27             ;REPORT REGISTER ERROR
5807
5808                                  ;RELEASE THE DRIVE FROM PORT A
5809
5810  033362  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5811  033370  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5812  033376  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5813
5814                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5815
5816  033404  005037  001242                  CLR     RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5817  033410  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
```

```
5818  033416  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5819  033422  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5820  033430  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5821  033436  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5822  033444  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5823  033452  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5824  033460  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5825  033466  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5826  033474  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5827  033502  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5828  033510  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5829  033516  001006                          BNE     66$             ;BR IF NOT
5830  033520  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5831  033524  001045                          BNE     68$             ;BR IF NOT
5832  033526  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5833  033530  000137  033714                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
5834  033534  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5835  033542  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5836  033550  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
5837  033556  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5838  033562  001414                          BEQ     67$             ;BR IF ZERO
5839  033564  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5840  033572  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5841  033600  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
5842  033606  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5843  033612  001012                          BNE     68$             ;BR IF NOT
5844  033614  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5845  033622  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
5846  033630  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
5847  033636  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5848  033640  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
5849  033646  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5850  033654  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5851  033662  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
5852  033664  104007                          ERROR   7               ;REPORT ERROR
5853  033666  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
5854  033674  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5855  033702  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5856  033710  001401                          BEQ     70$             ;BR IF OK
5857  033712  104007                          ERROR   7               ;REPORT ERROR
5858  033714  000240                  70$:    NOP
5859  033716                          1$:
5860
5861                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
5862
5863  033716  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
5864  033722  001412                          BEQ     TST26           ;;BR IF NOT
5865  033724  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
5866  033732  001406                          BEQ     TST26           ;;BR IF NOT
5867  033734  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
5868  033740  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
5869  033744  000177  145140                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
5870
5871                                  ;*****************************************************************
5872                                  ;*TEST 26       TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
5873                                  ;*
```

```
                                        ;*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 RHDS1.
                                        ;*
                                        ;*  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.
                                        ;*
                                        ;*****************************************************************
033750                                  TST26:
033750  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
033752  005737  001266                    :     TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
033756  001406                                  BEQ     2$              ;BR IF NOT
033760  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
033762  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
033766  012737  177777  001266    1$:           MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
033774  112737  000026  001102    2$:           MOVB    #26,$TSTNM      ;TEST NUMBER
034002  012737  034024  001106                  MOV     #TEST26,$LPADR  ;LOAD LOOP ON TEST ADDRESS
034010  012737  034024  001110                  MOV     #TEST26,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
034016  012737  007640  001170                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS

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

034024                                  TEST26:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

034024  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
034032  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
034036  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
034044  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
034052  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
034060  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
034064  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
034072  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;;*****************************************************************

                                                ;SEIZE THE DRIVE THROUGH PORT A

034100  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
034106  013737  001216  001230                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
034114  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
034120  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
034126  013737  001220  001226                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034134  013737  001220  001232                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
034142  016037  000012  001126                  MOV     RHDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
```

```
5930  034150  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
5931  034154  062737  000012  001122          ADD     #RHDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
5932  034162  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
5933  034166  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
5934  034174  001403                          BEQ     .+10            ;BR IF IT IS
5935  034176  104004                          ERROR   4               ;REPORT THE ERROR
5936  034200  000137  035132                  JMP     1$              ;BYPASS REST OF THE SUBTEST
5937  034204  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5938  034212  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5939  034220  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
5940  034226  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
5941  034234  013737  001124  001160          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
5942  034242  005137  001160                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
5943  034246  013737  001126  001156          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
5944  034254  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
5945  034262  023737  001124  001156          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
5946  034270  001401                          BEQ     .+4             ;BR IF THEY ARE
5947  034272  104005                          ERROR 5                 ;REPORT THE ERROR
5948
5949                                  ;;**************************************************************************
5950                                  ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
5951
5952  034274  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5953  034302  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5954  034310  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT B
5955
5956                                  ;;**************************************************************************
5957                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
5958
5959  034316  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5960  034322  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
5961  034330  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5962  034336  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5963  034342  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5964  034346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
5965  034354  001403                          BEQ     64$             ;BR IF OK
5966  034356  104010                          ERROR   10              ;REPORT THE ERROR
5967  034360  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5968  034364  016037  000000  001126  64$:    MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
5969  034372  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
5970  034400  001404                          BEQ     .+12            ;BR IF NOT
5971  034402  104011                          ERROR   11              ;REPORT THE ERROR
5972  034404  012760  040000  000000          MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
5973  034412  005737  001236                  TST     CKERR           ;WAS RHDS1 NON ZERO ?
5974  034416  001402                          BEQ     .+6             ;CONTENTS OF RHDS1 SEEN BY PORT B
5975  034420  000137  035132                  JMP     1$              ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5976
5977                                  ;;**************************************************************************
5978
5979                                          ;RELEASE THE DRIVE FROM PORT A
5980
5981  034424  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5982  034432  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5983  034440  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5984
5985                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
```

```
5986
5987  034446  005037  001242                    CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5988  034452  012737  111700  001124            MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5989  034460  012737  000012  001122            MOV     #RHDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
5990  034466  060037  001122                    ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5991  034472  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
5992  034500  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5993  034506  016037  000012  001156            MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
5994  034514  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
5995  034522  013737  001216  001226            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5996  034530  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
5997  034536  001404                            BEQ     65$             ;BR IF STATUS FROM PORT A ZERO
5998  034540  005737  001156                    TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
5999  034544  001401                            BEQ     65$             ;BR IF ZERO
6000  034546  104031                            ERROR   31              ;REPORT DRIVE IN NEUTRAL
6001  034550  013737  001156  001126  65$:     MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
6002  034556  013737  001220  001226            MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
6003  034564  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
6004  034572  001401                            BEQ     .+4             ;BR IF OK
6005  034574  104027                            ERROR   27              ;REPORT REGISTER ERROR
6006
6007                                            ;RELEASE THE DRIVE FROM PORT B
6008
6009  034576  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6010  034604  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6011  034612  012760  000013  000000            MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6012
6013                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6014
6015  034620  005037  001242                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6016  034624  012737  000012  001122            MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6017  034632  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6018  034636  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6019  034644  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6020  034652  016037  000012  001162            MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6021  034660  013737  001162  001156            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6022  034666  042737  100100  001156            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6023  034674  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6024  034702  016037  000012  001164            MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6025  034710  013737  001164  001160            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6026  034716  042737  100100  001160            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6027  034724  023737  001156  001160            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6028  034732  001006                            BNE     66$             ;BR IF NOT
6029  034734  005737  001156                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6030  034740  001045                            BNE     68$             ;BR IF NOT
6031  034742  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6032  034744  000137  035130                    JMP     70$             ;BYPASS THE REST OF THE CHECKS
6033  034750  013737  001162  001126  66$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6034  034756  013737  001220  001226            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6035  034764  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6036  034772  005737  001156                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6037  034776  001414                            BEQ     67$             ;BR IF ZERO
6038  035000  013737  001216  001226            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6039  035006  013737  001164  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6040  035014  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6041  035022  005737  001160                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
```

```
6042  035026  001012                        BNE     68$             ;BR IF NOT
6043  035030  012737  177777  001242  67$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6044  035036  012760  000011  000000        MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6045  035044  012760  000013  000000        MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6046  035052  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
6047  035054  013737  001162  001126  68$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6048  035062  013737  001216  001226        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6049  035070  023737  001124  001162        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6050  035076  001401                        BEQ     69$             ;BR IF OK FROM PORT A.
6051  035100  104007                        ERROR   7               ;REPORT ERROR
6052  035102  013737  001164  001126  69$:  MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6053  035110  013737  001220  001226        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6054  035116  023737  001124  001164        CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6055  035124  001401                        BEQ     70$             ;BR IF OK
6056  035126  104007                        ERROR   7               ;REPORT ERROR
6057  035130  000240                  70$:  NOP
6058  035132                          1$:
6059
6060                                        ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
6061
6062  035132  105737  001103                TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
6063  035136  001412                        BEQ     TST27           ;;BR IF NOT
6064  035140  032737  001000  177570        BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
6065  035146  001406                        BEQ     TST27           ;;BR IF NOT
6066  035150  105037  001103                CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
6067  035154  005037  001170                CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
6068  035160  000177  143724                JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
6069
6070
6071                                        ;*****************************************************************
6072                                        ;*TEST 27       TEST SEIZE BY WRITING ATTENTION BIT
6073                                        ;*
6074                                        ;*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
6075                                        ;*      (RHAS) SEIZES THE DRIVE.  VERIFY THAT REQUEST IS SET FOR THE OTHER
6076                                        ;*      PORT.
6077                                        ;*
6078                                        ;*  A.  WRITE THE APPROPRIATE DRIVE BIT INTO RHAS; VERIFY THAT THE DRIVE
6079                                        ;*      IS SEIZED.
6080                                        ;*
6081                                        ;*  B.  ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
6082                                        ;*      DRIVE SWITCHES TO THE OPPOSITE PORT.  ISSUE A RELEASE THRCUGH THE
6083                                        ;*      OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
6084                                        ;*
6085                                        ;*****************************************************************
6086  035164                          TST27:
6087  035164  000004                        SCOPE                   ;INITIALIZE THE SCOPE HANDLER
6088  035166  005737  001266                TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6089  035172  001406                        BEQ     2$              ;BR IF NOT
6090  035174  100002                        BPL     1$              ;BR IF JUST ENTERED TEST
6091  035176  000137  002410                JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6092  035202  012737  177777  001266  1$:  MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6093  035210  112737  000027  001102  2$:  MOVB    #27,$TSTNM      ;TEST NUMBER
6094  035216  012737  035240  001106        MOV     #TEST27,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6095  035224  012737  035240  001110        MOV     #TEST27,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6096  035232  012737  007640  001170        MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
6097
```

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

035240                                  TEST27:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

035240  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
035246  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
035252  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
035260  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
035266  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
035274  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
035300  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
035306  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

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

035314  113760  001222  000010                  MOVB    PORTC,RHCS2(R0) ;SELECT DRIVE NOT BEING TESTED
035322  013737  001216  001230                  MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS

                                        ;;*****************************************************************************
                                        ;WRITE THE DRIVE'S ATTENTION BIT

035330  013760  001224  000016                  MOV     ASR1,RHAS(R0)   ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
035336  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
035344  013737  001216  001226                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

035352  005760  000012                          TST     RHDS1(R0)       ;SEE THE REGISTER THROUGH PORT A ?
035356  001014                                  BNE     1$              ;BR IF YES
035360  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
035366  013737  001220  001226                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035374  005760  000012                          TST     RHDS1(R0)       ;SEE REGISTER THROUGH PORT B ?
035400  001021                                  BNE     2$              ;BR IF YES
035402  104042                                  ERROR   42              ;DRIVE NOT IN NEUTRAL OR SEIZED
035404  000137  037150                          JMP     4$              ;BYPASS REST OF TEST
035410                                  1$:
035410  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
035416  013737  001220  001226                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035424  005760  000012                          TST     RHDS1(R0)       ;REGISTER SHOULD BE ZERO THROUGH PORT B
035430  001002                                  BNE     .+6             ;BR IF STATUS REG IS NOT ZERO
035432  000137  036302                          JMP     3$              ;STATUS REG IS ZERO
035436  104043                                  ERROR   43              ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
035440  000137  037150                          JMP     4$              ;BYPASS REST OF TEST

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

035444                                  2$:
035444  005037  001236                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
035450  016037  000012  001126                  MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
```

```
6154  035456  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6155  035464  060037  001122                  ADD     R0,$BDADR      ;ADD RH11 BASE ADDRESS
6156  035470  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
6157  035476  013737  001126  001156          MOV     $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
6158  035504  042737  106077  001156          BIC     #^C71700,$TMP0 ;SAVE SPECIFIED BITS
6159  035512  023737  001124  001156          CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
6160  035520  001414                          BEQ     64$            ;BR IF OK
6161  035522  013737  001126  001166          MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
6162  035530  042737  071700  001166          BIC     #71700,$TMP4   ;CLEAR THE MASKED BITS
6163  035536  053737  001166  001124          BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
6164  035544  104010                          ERROR   10             ;REPORT THE ERROR
6165  035546  005137  001236                  COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
6166  035552  000240                  64$:    NOP
6167  035554  013737  001220  001230          MOV     PORTB,SEIZPT   ;ADDRESS FOR ERROR MESSAGE
6168  035562  013737  001216  001232          MOV     PORTA,OPPRT    ;SAME AS ABOVE
6169
6170                                          ;RELEASE THE DRIVE FROM PORT B
6171
6172  035570  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6173  035576  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6174  035604  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6175
6176                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
6177
6178  035612  005037  001242                  CLR     RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
6179  035616  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6180  035624  012737  000012  001122          MOV     #RHDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
6181  035632  060037  001122                  ADD     R0,$BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
6182  035636  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6183  035644  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6184  035652  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
6185  035660  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6186  035666  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6187  035674  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
6188  035702  001404                          BEQ     65$            ;BR IF STATUS FROM PORT B ZERO
6189  035704  005737  001156                  TST     $TMP0          ;IS STATUS FROM PORT A ZERO ?
6190  035710  001401                          BEQ     65$            ;BR IF ZERO
6191  035712  104044                          ERROR   44             ;REPORT DRIVE NOT SEIZED BY PORT A
6192  035714  013737  001156  001126  65$:    MOV     $TMP0,$BDDAT   ;CHECK STATUS FROM PORT A
6193  035722  013737  001216  001226          MOV     PORTA,PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
6194  035730  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
6195  035736  001401                          BEQ     .+4            ;BR IF OK
6196  035740  104027                          ERROR   27             ;REPORT REGISTER ERROR
6197
6198                                          ;RELEASE THE DRIVE FROM PORT A
6199
6200  035742  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6201  035750  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6202  035756  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
6203
6204                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6205
6206  035764  005037  001242                  CLR     RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6207  035770  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6208  035776  060037  001122                  ADD     R0,$BDADR      ;ADD THE I/O BASE ADDRESS
6209  036002  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
```

```
6210  036010  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6211  036016  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6212  036024  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6213  036032  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6214  036040  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6215  036046  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6216  036054  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6217  036062  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6218  036070  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6219  036076  001006                          BNE     66$             ;BR IF NOT
6220  036100  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6221  036104  001045                          BNE     68$             ;BR IF NOT
6222  036106  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6223  036110  000137  036274                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
6224  036114  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6225  036122  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6226  036130  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6227  036136  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6228  036142  001414                          BEQ     67$             ;BR IF ZERO
6229  036144  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6230  036152  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6231  036160  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6232  036166  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6233  036172  001012                          BNE     68$             ;BR IF NOT
6234  036174  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6235  036202  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6236  036210  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6237  036216  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6238  036220  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6239  036226  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6240  036234  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6241  036242  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
6242  036244  104007                          ERROR   7               ;REPORT ERROR
6243  036246  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6244  036254  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6245  036262  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6246  036270  001401                          BEQ     70$             ;BR IF OK
6247  036272  104007                          ERROR   7               ;REPORT ERROR
6248  036274  000240                  70$:    NOP
6249  036276  000137  037150                  JMP     4$
6250
6251                                          ;;*************************************************************************
6252                                          ;THE DRIVE IS SEIZED BY PORT A.  VERIFY THAT PORT B HAS PORT REQUEST SET
6253
6254  036302                          3$:
6255  036302  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6256  036310  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6257  036316  005037  001236                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
6258  036322  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
6259  036330  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6260  036336  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
6261  036342  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
6262  036350  013737  001126  001156          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
6263  036356  042737  106077  001156          BIC     #^C71700,$TMP0   ;SAVE SPECIFIED BITS
6264  036364  023737  001124  001156          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
6265  036372  001414                          BEQ     71$              ;BR IF OK
```

```
6266  036374  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6267  036402  042737  071700  001166          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
6268  036410  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6269  036416  104010                          ERROR   10              ;REPORT THE ERROR
6270  036420  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6271  036424  000240                  71$:    NOP
6272  036426  013737  001216  001230          MOV     PORTA,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
6273  036434  013737  001220  001232          MOV     PORTB,OPPRT     ;SAME AS ABOVE
6274
6275                                          ;RELEASE THE DRIVE FROM PORT A
6276
6277  036442  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6278  036450  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6279  036456  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
6280
6281                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
6282
6283  036464  005037  001242                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
6284  036470  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6285  036476  012737  000012  001122          MOV     #RHDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
6286  036504  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
6287  036510  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6288  036516  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6289  036524  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
6290  036532  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6291  036540  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6292  036546  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
6293  036554  001404                          BEQ     72$             ;BR IF STATUS FROM PORT A ZERO
6294  036556  005737  001156                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
6295  036562  001401                          BEQ     72$             ;BR IF ZERO
6296  036564  104044                          ERROR   44              ;REPORT DRIVE NOT SEIZED BY PORT B
6297  036566  013737  001156  001126  72$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
6298  036574  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
6299  036602  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
6300  036610  001401                          BEQ     .+4             ;BR IF OK
6301  036612  104027                          ERROR   27              ;REPORT REGISTER ERROR
6302
6303                                          ;RELEASE THE DRIVE FROM PORT B
6304
6305  036614  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6306  036622  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6307  036630  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6308
6309                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6310
6311  036636  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6312  036642  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6313  036650  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6314  036654  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6315  036662  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6316  036670  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6317  036676  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6318  036704  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6319  036712  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6320  036720  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6321  036726  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
```

```
6322  036734  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6323  036742  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6324  036750  001006                          BNE     73$             ;BR IF NOT
6325  036752  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6326  036756  001045                          BNE     75$             ;BR IF NOT
6327  036760  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6328  036762  000137  037146                  JMP     77$             ;BYPASS THE REST OF THE CHECKS
6329  036766  013737  001162  001126  73$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6330  036774  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6331  037002  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6332  037010  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6333  037014  001414                          BEQ     74$             ;BR IF ZERO
6334  037016  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6335  037024  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6336  037032  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6337  037040  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6338  037044  001012                          BNE     75$             ;BR IF NOT
6339  037046  012737  177777  001242  74$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6340  037054  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6341  037062  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6342  037070  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6343  037072  013737  001162  001126  75$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6344  037100  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6345  037106  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6346  037114  001401                          BEQ     76$             ;BR IF OK FROM PORT A.
6347  037116  104007                          ERROR   7               ;REPORT ERROR
6348  037120  013737  001164  001126  76$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6349  037126  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6350  037134  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6351  037142  001401                          BEQ     77$             ;BR IF OK
6352  037144  104007                          ERROR   7               ;REPORT ERROR
6353  037146  000240                  77$:    NOP                     .
6354  037150                          4$:                             .
6355
6356                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
6357
6358  037150  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
6359  037154  001412                          BEQ     TST30           ;;BR IF NOT
6360  037156  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
6361  037164  001406                          BEQ     TST30           ;;BR IF NOT
6362  037166  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
6363  037172  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
6364  037176  000177  141706                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
6365
6366
6367                                  ;*****************************************************************
6368                                  ;*TEST 30       TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
6369                                  ;*
6370                                  ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
6371                                  ;*      THE DRIVE'S ATTENTION BIT.
6372                                  ;*
6373                                  ;*  A.  SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
6374                                  ;*      BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
6375                                  ;*
6376                                  ;*  B.  VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
6377                                  ;*
```

```
6378                                    ;*****************************************************************
6379  037202                            TST30:
6380  037202  000004                            SCOPE                   ;INITIALIZE THE SCOPE HANDLER
6381  037204  005737  001266                    TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6382  037210  001406                            BEQ     2$              ;BR IF NOT
6383  037212  100002                            BPL     1$              ;BR IF JUST ENTERED TEST
6384  037214  000137  002410                    JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6385  037220  012737  177777  001266  1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6386  037226  112737  000030  001102  2$:       MOVB    #30,$TSTNM      ;TEST NUMBER
6387  037234  012737  037256  001106            MOV     #TEST30,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6388  037242  012737  037256  001110            MOV     #TEST30,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6389  037250  012737  007640  001170            MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
6390
6391
6392                                    ;;*****************************************************************************
6393                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
6394
6395  037256                            TEST30:
6396
6397                                            ;CLEAR ATTENTION BITS FOR BOTH PORTS
6398
6399  037256  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
6400  037264  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE
6401  037270  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
6402  037276  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6403  037304  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
6404  037312  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
6405  037316  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
6406  037324  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6407  037332  113760  001222  000010            MOVB    PORTC,RHCS2(R0) ;SELECT DRIVE NOT BEING TESTED
6408
6409                                    ;;*****************************************************************************
6410                                    ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST
6411
6412  037340  013737  001224  001156            MOV     ASR1,$TMP0      ;STORE ATTN BIT FOR PORT A
6413  037346  005137  001156                    COM     $TMP0           ;COMPLEMENT IT
6414  037352  013760  001156  000016            MOV     $TMP0,RHAS(R0)  ;WRITE THE ATTN REGISTER
6415
6416                                    ;;*****************************************************************************
6417                                    ;VERIFY THAT DRIVE REMAINDED IN NEUTRAL
6418
6419
6420                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6421
6422  037360  005037  001242                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6423  037364  012737  000012  001122            MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6424  037372  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6425  037376  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6426  037404  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6427  037412  016037  000012  001162            MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6428  037420  013737  001162  001156            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6429  037426  042737  100100  001156            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6430  037434  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6431  037442  016037  000012  001164            MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6432  037450  013737  001164  001160            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6433  037456  042737  100100  001160            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```

```
6434  037464  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6435  037472  001006                          BNE     64$             ;BR IF NOT
6436  037474  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6437  037500  001045                          BNE     66$             ;BR IF NOT
6438  037502  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6439  037504  000137  037670                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
6440  037510  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6441  037516  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6442  037524  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6443  037532  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6444  037536  001414                          BEQ     65$             ;BR IF ZERO
6445  037540  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6446  037546  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6447  037554  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6448  037562  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6449  037566  001012                          BNE     66$             ;BR IF NOT
6450  037570  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6451  037576  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6452  037604  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6453  037612  104021                          ERROR   21              ;TYPE ERROR MESSAGE 21
6454  037614  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6455  037622  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6456  037630  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6457  037636  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
6458  037640  104007                          ERROR   7               ;REPORT ERROR
6459  037642  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6460  037650  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6461  037656  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6462  037664  001401                          BEQ     68$             ;BR IF OK
6463  037666  104007                          ERROR   7               ;REPORT ERROR
6464  037670  000240                  68$:    NOP
6465
6466                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
6467
6468  037672  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
6469  037676  001412                          BEQ     TST31           ;;BR IF NOT
6470  037700  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
6471  037706  001406                          BEQ     TST31           ;;BR IF NOT
6472  037710  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
6473  037714  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
6474  037720  000177  141164                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
6475
6476
6477                                  ;*****************************************************************
6478                                  ;*TEST 31       TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
6479                                  ;*
6480                                  ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
6481                                  ;*
6482                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
6483                                  ;*
6484                                  ;*  B.  WRITE 1'S INTO RHER1 THROUGH PORT 'A'.
6485                                  ;*
6486                                  ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
6487                                  ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR
6488                                  ;*      PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
6489                                  ;*
```

```
                                        ;*****************************************************************
037724                                  TST31:
037724  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
037726  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
037732  001406                                  BEQ     2$              ;BR IF NOT
037734  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
037736  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
037742  012737  177777  001266  1$:     MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
037750  112737  000031  001102  2$:     MOVB    #31,$TSTNM              ;TEST NUMBER
037756  012737  040000  001106          MOV     #TEST31,$LPADR          ;LOAD LOOP ON TEST ADDRESS
037764  012737  040000  001110          MOV     #TEST31,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
037772  012737  000004  001170          MOV     #4,$TIMES               ;;DO 4 ITERATIONS

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

040000                                  TEST31:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

040000  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
040006  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
040012  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
040020  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
040026  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
040034  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
040040  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
040046  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
                                        ;;*****************************************************************

                                                ;SEIZE THE DRIVE THROUGH PORT A

040054  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
040062  013737  001216  001230                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
040070  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
040074  013737  001220  001232                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;;*****************************************************************
                                        ;FORCE AN ERROR

040102  012760  177777  000014                  MOV     #-1,RHER1(R0)   ;SET ERROR BITS

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

040110  005037  001244                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
040114  012737  003720  001246                  MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
040122  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
040130  013737  001220  001226                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;*****************************************************************
                                        ;WAIT FOR DRIVE TO TIMEOUT

040136  005760  000012          1$:     TST     RHDS1(R0)               ;WAIT FOR THE DRIVE TO BE RELEASED
040142  001004                          BNE     2$                      ;BR IF DRIVE RELEASED
```

```
6546  040144  005737  001246          TST     WATCH           ;WATCH AT ZERO ?
6547  040150  001372                  BNE     1$              ;BR IF NOT
6548  040152  104036                  ERROR   36              ;DRIVE NOT RELEASED WITHIN 2 SECONDS
6549  040154                  2$:
6550  040154  113760  001216  000010  MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6551  040162  013737  001216  001226  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6552
6553                  ;;**************************************************************************
6554                  ;THE ERROR BIT ('ERR') IN RHDS1 SHOULD STILL BE SET
6555
6556  040170  005037  001236          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6557  040174  016037  000012  001126  MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
6558  040202  012737  000012  001122  MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6559  040210  060037  001122          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6560  040214  012737  040000  001124  MOV     #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
6561  040222  013737  001126  001156  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6562  040230  042737  137777  001156  BIC     #^C40000,$TMP0  ;SAVE SPECIFIED BITS
6563  040236  023737  001124  001156  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6564  040244  001414                  BEQ     64$             ;BR IF OK
6565  040246  013737  001126  001166  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6566  040254  042737  040000  001166  BIC     #40000,$TMP4    ;CLEAR THE MASKED BITS
6567  040262  053737  001166  001124  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6568  040270  104023                  ERROR   23              ;TYPE MESSAGE 23
6569  040272  005137  001236          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6570  040276  000240          64$:    NOP
6571
6572                  ;;**************************************************************************
6573                  ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6574
6575  040300  005037  001236          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6576  040304  016037  000014  001126  MOV     RHER1(R0),$BDDAT  ;GET CONTENTS OF RHER1
6577  040312  012737  000014  001122  MOV     #RHER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6578  040320  060037  001122          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6579  040324  012737  177777  001124  MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
6580  040332  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
6581  040340  001403                  BEQ     65$             ;BR IF OK
6582  040342  104010                  ERROR   10              ;REPORT THE ERROR
6583  040344  005137  001236          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6584  040350  000240          65$:    NOP
6585
6586                  ;;**************************************************************************
6587                  ;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET
6588
6589  040352  005037  001236          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6590  040356  016037  000012  001126  MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
6591  040364  012737  000012  001122  MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6592  040372  060037  001122          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6593  040376  012737  100000  001124  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
6594  040404  013737  001126  001156  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6595  040412  042737  077777  001156  BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
6596  040420  023737  001124  001156  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6597  040426  001414                  BEQ     66$             ;BR IF OK
6598  040430  013737  001126  001166  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6599  040436  042737  100000  001166  BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6600  040444  053737  001166  001124  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6601  040452  104041                  ERROR   41              ;TYPE MESSAGE 41
```

```
6602  040454  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6603  040460  000240                  66$:    NOP
6604
6605
6606                                  ;;**************************************************************************
6607
6608                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6609
6610  040462  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6611  040466  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6612  040474  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6613  040500  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
6614  040506  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6615  040514  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6616  040522  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6617  040530  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6618  040536  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6619  040544  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6620  040552  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6621  040560  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6622  040566  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6623  040574  001006                          BNE     67$             ;BR IF NOT
6624  040576  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6625  040602  001045                          BNE     69$             ;BR IF NOT
6626  040604  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6627  040606  000137  041006                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
6628  040612  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6629  040620  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6630  040626  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6631  040634  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6632  040640  001414                          BEQ     68$             ;BR IF ZERO
6633  040642  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6634  040650  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6635  040656  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6636  040664  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6637  040670  001012                          BNE     69$             ;BR IF NOT
6638  040672  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6639  040700  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6640  040706  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6641  040714  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6642  040716  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6643  040724  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6644  040732  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
6645  040740  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6646  040746  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
6647  040750  104007                          ERROR   7               ;REPORT ERROR
6648  040752  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6649  040760  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6650  040766  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6651  040774  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6652  041002  001401                          BEQ     71$             ;BR IF OK
6653  041004  104007                          ERROR   7               ;REPORT ERROR
6654  041006  000240                  71$:    NOP
6655
6656                                  ;;**************************************************************************
6657                                  ;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET
```

```
6658
6659  041010  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
6660  041016  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6661  041024  005037  001236                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
6662  041030  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
6663  041036  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6664  041044  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
6665  041050  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
6666  041054  013737  001126  001156          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
6667  041062  042737  077777  001156          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
6668  041070  023737  001124  001156          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
6669  041076  001414                          BEQ     72$              ;BR IF OK
6670  041100  013737  001126  001166          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
6671  041106  042737  100000  001166          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
6672  041114  053737  001166  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
6673  041122  104052                          ERROR   52               ;TYPE MESSAGE 52
6674  041124  005137  001236                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
6675  041130  000240                  72$:    NOP
6676
6677                                          ;CLEAR ATTENTION BIT FOR PORT A
6678
6679  041132  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT #A
6680  041140  005060  000012                  CLR     RHDS1(R0)        ;SEIZE THE DRIVE
6681  041144  012760  000011  000000          MOV     #11,RHCS1(R0)    ;ISSUE DRIVE CLEAR
6682  041152  012760  000013  000000          MOV     #13,RHCS1(R0)    ;RELEASE THE DRIVE
6683  041160                          3$:
6684
6685                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
6686
6687  041160  105737  001103                  TSTB    $ERFLG           ;DID AN ERROR OCCUR ?
6688  041164  001412                          BEQ     TST32            ;;BR IF NOT
6689  041166  032737  001000  177570          BIT     #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
6690  041174  001406                          BEQ     TST32            ;;BR IF NOT
6691  041176  105037  001103                  CLRB    $ERFLG           ;CLEAR THE ERROR FLAG
6692  041202  005037  001170                  CLR     $TIMES           ;CLEAR THE MAX ITERATION COUNT
6693  041206  000177  137676                  JMP     @$LPERR          ;GO TO THE LOOP ADDRESS
6694
6695                                  ;*****************************************************************
6696                                  ;*TEST 32       TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
6697                                  ;*
6698                                  ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
6699                                  ;*
6700                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
6701                                  ;*
6702                                  ;*  B.  WRITE 1'S INTO RHER1 THROUGH PORT 'B'.
6703                                  ;*
6704                                  ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
6705                                  ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
6706                                  ;*      PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
6707                                  ;*
6708                                  ;*****************************************************************
6709  041212                  TST32:
6710  041212  000004                          SCOPE                    ;INITIALIZE THE SCOPE HANDLER
6711  041214  005737  001266                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TESTS ?
6712  041220  001406                          BEQ     2$               ;BR IF NOT
6713  041222  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
```

```
                                                JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
                                1$:             MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
                                2$:             MOVB    #32,$TSTNM      ;TEST NUMBER
                                                MOV     #TEST32,$LPADR  ;LOAD LOOP ON TEST ADDRESS
                                                MOV     #TEST32,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
                                                MOV     #4,$TIMES       ;;DO 4 ITERATIONS

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

041266                          TEST32:

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

                                                MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
                                                CLR     RHDS1(R0)       ;SEIZE THE DRIVE
                                                MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
                                                MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
                                                MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
                                                CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
                                                MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
                                                MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
                                ;;**************************************************************************

                                        ;SEIZE THE DRIVE THROUGH PORT B

                                                MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
                                                MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
                                                CLR     RHDS1(R0)       ;WRITE RHDS1
                                                MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                ;;**************************************************************************
                                ;FORCE AN ERROR

                                                MOV     #-1,RHER1(R0)   ;SET ERROR BITS

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

                                                CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
                                                MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
                                                MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
                                                MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;;**************************************************************************
                                ;WAIT FOR DRIVE TO TIMEOUT

                                1$:             TST     RHDS1(R0)       ;WAIT FOR THE DRIVE TO BE RELEASED
                                                BNE     2$              ;BR IF DRIVE RELEASED
                                                TST     WATCH           ;WATCH AT ZERO ?
                                                BNE     1$              ;BR IF NOT
                                                ERROR   36              ;DRIVE NOT RELEASED WITHIN 2 SECONDS
                                2$:
                                                MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
                                                MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
6770
6771                                    ;;*************************************************************************
6772                                    ;THE ERROR BIT ('ERR') IN RHDS1 SHOULD STILL BE SET
6773
6774  041456  005037  001236                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
6775  041462  016037  000012  001126          MOV     RHDS1(R0),$BDDAT    ;GET CONTENTS OF RHDS1
6776  041470  012737  000012  001122          MOV     #RHDS1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6777  041476  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
6778  041502  012737  040000  001124          MOV     #ERR,$GDDAT         ;WHAT REGISTER SHOULD BE
6779  041510  013737  001126  001156          MOV     $BDDAT,$TMP0        ;MOVE REGISTER CONTENTS TO '$TMP0'
6780  041516  042737  137777  001156          BIC     #^C40000,$TMP0      ;SAVE SPECIFIED BITS
6781  041524  023737  001124  001156          CMP     $GDDAT,$TMP0        ;COMPARE THE BITS
6782  041532  001414                          BEQ     64$                 ;BR IF OK
6783  041534  013737  001126  001166          MOV     $BDDAT,$TMP4        ;COPY 'BAD DATA'
6784  041542  042737  040000  001166          BIC     #40000,$TMP4        ;CLEAR THE MASKED BITS
6785  041550  053737  001166  001124          BIS     $TMP4,$GDDAT        ;'OR' WITH GOOD DATA FOR TYPEOUT
6786  041556  104023                          ERROR   23                  ;TYPE MESSAGE 23
6787  041560  005137  001236                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6788  041564  000240                64$:      NOP
6789
6790                                    ;;*************************************************************************
6791                                    ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6792
6793  041566  005037  001236                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
6794  041572  016037  000014  001126          MOV     RHER1(R0),$BDDAT    ;GET CONTENTS OF RHER1
6795  041600  012737  000014  001122          MOV     #RHER1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6796  041606  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
6797  041612  012737  177777  001124          MOV     #177777,$GDDAT      ;WHAT REGISTER SHOULD BE
6798  041620  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;IS THE REGISTER OK ?
6799  041626  001403                          BEQ     65$                 ;BR IF OK
6800  041630  104010                          ERROR   10                  ;REPORT THE ERROR
6801  041632  005137  001236                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6802  041636  000240                65$:      NOP
6803
6804                                    ;;*************************************************************************
6805                                    ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
6806
6807  041640  005037  001236                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
6808  041644  016037  000012  001126          MOV     RHDS1(R0),$BDDAT    ;GET CONTENTS OF RHDS1
6809  041652  012737  000012  001122          MOV     #RHDS1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6810  041660  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
6811  041664  012737  100000  001124          MOV     #ATA,$GDDAT         ;WHAT REGISTER SHOULD BE
6812  041672  013737  001126  001156          MOV     $BDDAT,$TMP0        ;MOVE REGISTER CONTENTS TO '$TMP0'
6813  041700  042737  077777  001156          BIC     #^CATA,$TMP0        ;SAVE SPECIFIED BITS
6814  041706  023737  001124  001156          CMP     $GDDAT,$TMP0        ;COMPARE THE BITS
6815  041714  001414                          BEQ     66$                 ;BR IF OK
6816  041716  013737  001126  001166          MOV     $BDDAT,$TMP4        ;COPY 'BAD DATA'
6817  041724  042737  100000  001166          BIC     #ATA,$TMP4          ;CLEAR THE MASKED BITS
6818  041732  053737  001166  001124          BIS     $TMP4,$GDDAT        ;'OR' WITH GOOD DATA FOR TYPEOUT
6819  041740  104041                          ERROR   41                  ;TYPE MESSAGE 41
6820  041742  005137  001236                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6821  041746  000240                66$:      NOP
6822
6823
6824                                    ;;*************************************************************************
6825
```

```
6826                                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6827
6828  041750  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6829  041754  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
6830  041762  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6831  041766  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
6832  041774  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6833  042002  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6834  042010  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6835  042016  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6836  042024  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6837  042032  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6838  042040  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6839  042046  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6840  042054  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6841  042062  001006                          BNE     67$             ;BR IF NOT
6842  042064  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6843  042070  001045                          BNE     69$             ;BR IF NOT
6844  042072  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6845  042074  000137  042274                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
6846  042100  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6847  042106  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6848  042114  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
6849  042122  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6850  042126  001414                          BEQ     68$             ;BR IF ZERO
6851  042130  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6852  042136  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6853  042144  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
6854  042152  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6855  042156  001012                          BNE     69$             ;BR IF NOT
6856  042160  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6857  042166  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
6858  042174  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6859  042202  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6860  042204  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
6861  042212  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6862  042220  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
6863  042226  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6864  042234  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
6865  042236  104007                          ERROR   7               ;REPORT ERROR
6866  042240  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
6867  042246  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6868  042254  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6869  042262  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6870  042270  001401                          BEQ     71$             ;BR IF OK
6871  042272  104007                          ERROR   7               ;REPORT ERROR
6872  042274  000240                  71$:    NOP
6873
6874                                          ;;**************************************************************************
6875                                          ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET
6876
6877  042276  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
6878  042304  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6879  042312  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6880  042316  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
6881  042324  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
```

```
6882  042332  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6883  042336  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6884  042342  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6885  042350  042737  077777  001156          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
6886  042356  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6887  042364  001414                          BEQ     72$             ;BR IF OK
6888  042366  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6889  042374  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6890  042402  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6891  042410  104052                          ERROR   52              ;TYPE MESSAGE 52
6892  042412  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6893  042416  000240                  72$:    NOP
6894
6895                                          ;CLEAR ATTENTION BIT FOR PORT B
6896
6897  042420  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT #B
6898  042426  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
6899  042432  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
6900  042440  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6901  042446                          3$:
6902
6903                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
6904
6905  042446  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
6906  042452  001412                          BEQ     TST33           ;;BR IF NOT
6907  042454  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
6908  042462  001406                          BEQ     TST33           ;;BR IF NOT
6909  042464  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
6910  042470  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
6911  042474  000177  136410                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
6912
6913
6914                                          ;*****************************************************************
6915                                          ;*TEST 33       TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
6916                                          ;*
6917                                          ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
6918                                          ;*      BITS ARE SET IN THE DRIVE.
6919                                          ;*
6920                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
6921                                          ;*
6922                                          ;*  B.  WRITE 1'S INTO RHER1 THROUGH PORT 'A'.
6923                                          ;*
6924                                          ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE 'GO'
6925                                          ;*      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
6926                                          ;*      THAT RHER1 HAS NOT BEEN CLEARED.
6927                                          ;*
6928                                          ;*  D.  CLEAR RHER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
6929                                          ;*
6930                                          ;*  E.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
6931                                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
6932                                          ;*
6933                                          ;*****************************************************************
6934  042500                          TST33:
6935  042500  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
6936  042502  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6937  042506  001406                          BEQ     2$              ;BR IF NOT
```

```
6938  042510  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
6939  042512  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6940  042516  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6941  042524  112737  000033  001102  2$:     MOVB    #33,$TSTNM      ;TEST NUMBER
6942  042532  012737  042554  001106          MOV     #TEST33,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6943  042540  012737  042554  001110          MOV     #TEST33,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6944  042546  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
6945
6946
6947                                  ;;*************************************************************************
6948                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
6949
6950  042554                          TEST33:
6951
6952
6953
6954  042554  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
6955  042562  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
6956  042566  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
6957  042574  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6958  042602  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
6959  042610  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
6960  042614  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
6961  042622  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
6962                                  ;;*************************************************************************
6963
6964                                          ;SEIZE THE DRIVE THROUGH PORT A
6965
6966  042630  113760  001216  0[illegible]10  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
6967  042636  013737  001216  001230          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
6968  042644  005060  000012                  CLR     RHDS1(R0)       ;WRITE RHDS1
6969  042650  013737  001220  001232          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
6970
6971                                  ;;*************************************************************************
6972                                  ;FORCE AN ERROR
6973
6974  042656  012760  177777  000014          MOV     #-1,RHER1(R0)   ;SET ERROR BITS
6975  042664  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND
6976  042672  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6977  042676  016037  000000  001126          MOV     RHCS1(R0),$BDDAT  ;GET CONTENTS OF RHCS1
6978  042704  012737  000000  001122          MOV     #RHCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6979  042712  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6980  042716  012737  004012  001124          MOV     #4012,$GDDAT    ;WHAT REGISTER SHOULD BE
6981  042724  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6982  042732  042737  173765  001156          BIC     #^C4012,$TMP0   ;SAVE SPECIFIED BITS
6983  042740  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6984  042746  001414                          BEQ     64$             ;BR IF OK
6985  042750  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6986  042756  042737  004012  001166          BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
6987  042764  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6988  042772  104025                          ERROR   25              ;TYPE MESSAGE 25
6989  042774  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6990  043000  000240                  64$:    NOP
6991  043002  005737  001236                  TST     CKERR           ;DID 'GO' BIT RESET ?
6992  043006  001002                          BNE     .+6             ;BR IF NOT
6993  043010  000137  043050                  JMP     1$              ;'GO' BIT RESET
```

```
6994  043014  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;INIT THE RH11
6995  043022  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
6996  043030  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6997  043036  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT A
6998  043044  000137  043564                  JMP     2$              ;BYPASS THE REST OF THE TEST
6999
7000                                  ;;**************************************************************************
7001                                  ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
7002
7003  043050                          1$:
7004  043050  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
7005  043056  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7006  043064  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7007  043070  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
7008  043076  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7009  043104  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7010  043110  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7011  043114  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7012  043122  001403                          BEQ     65$             ;BR IF OK
7013  043124  104024                          ERROR   24              ;TYPE MESSAGE 24
7014  043126  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7015  043132  000240                  65$:    NOP
7016  043134  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
7017  043142  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7018  043150  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7019  043154  016037  000014  001126          MOV     RHER1(R0),$BDDAT  ;GET CONTENTS OF RHER1
7020  043162  012737  000014  001122          MOV     #RHER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7021  043170  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7022  043174  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
7023  043202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7024  043210  001403                          BEQ     66$             ;BR IF OK
7025  043212  104010                          ERROR   10              ;REPORT THE ERROR
7026  043214  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7027  043220  000240                  66$:    NOP
7028
7029                                  ;;**************************************************************************
7030                                  ;CLEAR THE ERRORS THROUGH PORT A
7031
7032  043222  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE A DRIVE CLEAR
7033
7034                                  ;;**************************************************************************
7035
7036                                          ;RELEASE THE DRIVE FROM PORT A
7037
7038  043230  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
7039  043236  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7040  043244  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7041
7042                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7043
7044  043252  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7045  043256  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
7046  043264  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7047  043270  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7048  043276  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7049  043304  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
```

```
7050  043312  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7051  043320  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7052  043326  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7053  043334  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7054  043342  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7055  043350  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7056  043356  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7057  043364  001006                          BNE     67$             ;BR IF NOT
7058  043366  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7059  043372  001045                          BNE     69$             ;BR IF NOT
7060  043374  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7061  043376  000137  043562                  JMP     71$             ;BYPASS THE REST OF THE CHECKS
7062  043402  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7063  043410  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7064  043416  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7065  043424  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7066  043430  001414                          BEQ     68$             ;BR IF ZERO
7067  043432  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7068  043440  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7069  043446  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7070  043454  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7071  043460  001012                          BNE     69$             ;BR IF NOT
7072  043462  012737  177777  001242  68$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7073  043470  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
7074  043476  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7075  043504  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7076  043506  013737  001162  001126  69$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
7077  043514  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7078  043522  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7079  043530  001401                          BEQ     70$             ;BR IF OK FROM PORT A.
7080  043532  104007                          ERROR   7               ;REPORT ERROR
7081  043534  013737  001164  001126  70$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
7082  043542  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7083  043550  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7084  043556  001401                          BEQ     71$             ;BR IF OK
7085  043560  104007                          ERROR   7               ;REPORT ERROR
7086  043562  000240                  71$:    NOP
7087  043564                          2$:
7088
7089                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
7090
7091  043564  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
7092  043570  001412                          BEQ     TST34           ;;BR IF NOT
7093  043572  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
7094  043600  001406                          BEQ     TST34           ;;BR IF NOT
7095  043602  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
7096  043606  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
7097  043612  000177  135272                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
7098
7099                          ;*****************************************************************
7100                          ;*TEST 34       TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
7101                          ;*
7102                          ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
7103                          ;*      BITS ARE SET IN THE DRIVE.
7104                          ;*
7105                          ;*  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.
                                        ;*
                                        ;*****************************************************************
043616                                  TST34:
043616  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
043620  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
043624  001406                                  BEQ     2$              ;BR IF NOT
043626  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
043630  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
043634  012737  177777  001266  1$:     MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
043642  112737  000034  001102  2$:     MOVB    #34,$TSTNM              ;TEST NUMBER
043650  012737  043672  001106          MOV     #TEST34,$LPADR          ;LOAD LOOP ON TEST ADDRESS
043656  012737  043672  001110          MOV     #TEST34,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
043664  012737  007640  001170          MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS

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

043672                                  TEST34:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

043672  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
043700  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
043704  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
043712  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
043720  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
043726  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
043732  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
043740  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
                                        ;;*****************************************************************

                                                ;SEIZE THE DRIVE THROUGH PORT B

043746  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
043754  013737  001220  001230                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
043762  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
043766  013737  001216  001232                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;;*****************************************************************
                                        ;FORCE AN ERROR

043774  012760  177777  000014                  MOV     #-1,RHER1(R0)   ;SET ERROR BITS
044002  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;ISSUE A RELEASE COMMAND
044010  005037  001236                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
7162  044014  016037  000000  001126          MOV     RHCS1(R0),$BDDAT  ;GET CONTENTS OF RHCS1
7163  044022  012737  000000  001122          MOV     #RHCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7164  044030  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7165  044034  012737  004012  001124          MOV     #4012,$GDDAT  ;WHAT REGISTER SHOULD BE
7166  044042  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
7167  044050  042737  173765  001156          BIC     #^C4012,$TMP0  ;SAVE SPECIFIED BITS
7168  044056  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
7169  044064  001414                          BEQ     64$             ;BR IF OK
7170  044066  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
7171  044074  042737  004012  001166          BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
7172  044102  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
7173  044110  104025                          ERROR   25              ;TYPE MESSAGE 25
7174  044112  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7175  044116  000240                  64$:    NOP
7176  044120  005737  001236                  TST     CKERR           ;DID 'GO' BIT RESET ?
7177  044124  001002                          BNE     .+6             ;BR IF NOT
7178  044126  000137  044166                  JMP     1$              ;'GO' BIT RESET
7179  044132  012760  000040  000010          MOV     #CLR,RHCS2(R0)  ;INIT THE RH11
7180  044140  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
7181  044146  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7182  044154  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT B
7183  044162  000137  044702                  JMP     2$              ;BYPASS THE REST OF THE TEST
7184
7185                          ;;*************************************************************************
7186                          ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
7187
7188  044166                  1$:
7189  044166  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
7190  044174  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7191  044202  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7192  044206  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
7193  044214  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7194  044222  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7195  044226  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7196  044232  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7197  044240  001403                          BEQ     65$             ;BR IF OK
7198  044242  104024                          ERROR   24              ;TYPE MESSAGE 24
7199  044244  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7200  044250  000240                  65$:    NOP
7201  044252  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
7202  044260  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7203  044266  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7204  044272  016037  000014  001126          MOV     RHER1(R0),$BDDAT  ;GET CONTENTS OF RHER1
7205  044300  012737  000014  001122          MOV     #RHER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7206  044306  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7207  044312  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
7208  044320  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7209  044326  001403                          BEQ     66$             ;BR IF OK
7210  044330  104010                          ERROR   10              ;REPORT THE ERROR
7211  044332  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7212  044336  000240                  66$:    NOP
7213
7214                          ;;*************************************************************************
7215                          ;CLEAR THE ERRORS THROUGH PORT B
7216
7217  044340  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE A DRIVE CLEAR
```

```
                                ;;*****************************************************************************
                                ;RELEASE THE DRIVE FROM PORT B
                                ;
044346  113760  001220  000010        MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
044354  013737  001220  001226        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044362  012760  000013  000000        MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
                                ;
                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
                                ;
044370  005037  001242                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
044374  012737  000012  001122        MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
044402  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
044406  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
044414  113760  001216  000010        MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
044422  016037  000012  001162        MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044430  013737  001162  001156        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
044436  042737  100100  001156        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044444  113760  001220  000010        MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
044452  016037  000012  001164        MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
044460  013737  001164  001160        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
044466  042737  100100  001160        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044474  023737  001156  001160        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
044502  001006                        BNE     67$             ;BR IF NOT
044504  005737  001156                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
044510  001045                        BNE     69$             ;BR IF NOT
044512  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
044514  000137  044700                JMP     71$             ;BYPASS THE REST OF THE CHECKS
044520  013737  001162  001126  67$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
044526  013737  001220  001226        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
044534  113760  001220  000010        MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
044542  005737  001156                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
044546  001414                        BEQ     68$             ;BR IF ZERO
044550  013737  001216  001226        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
044556  013737  001164  001126        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
044564  113760  001216  000010        MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
044572  005737  001160                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
044576  001012                        BNE     69$             ;BR IF NOT
044600  012737  177777  001242  68$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
044606  012760  000011  000000        MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
044614  012760  000013  000000        MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
044622  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
044624  013737  001162  001126  69$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
044632  013737  001216  001226        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
044640  023737  001124  001162        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
044646  001401                        BEQ     70$             ;BR IF OK FROM PORT A.
044650  104007                        ERROR   7               ;REPORT ERROR
044652  013737  001164  001126  70$:  MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
044660  013737  001220  001226        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
044666  023737  001124  001164        CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
044674  001401                        BEQ     71$             ;BR IF OK
044676  104007                        ERROR   7               ;REPORT ERROR
044700  000240                71$:    NOP
044702                        2$:
```

```
                                        ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

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

                                ;*****************************************************************
                                ;*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 + OR - 25% OF THE SPECIFIED
                                ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
                                ;*
                                ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
                                ;*      BIT IS SET.
                                ;*
                                ;*****************************************************************
044734                          TST35:
044734  000004                            SCOPE                   ;INITIALIZE THE SCOPE HANDLER
044736  005737  001266                    TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
044742  001406                            BEQ     2$              ;BR IF NOT
044744  100002                            BPL     1$              ;BR IF JUST ENTERED TEST
044746  000137  002410                    JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
044752  012737  177777  001266  1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
044760  112737  000035  001102  2$:       MOVB    #35,$TSTNM      ;TEST NUMBER
044766  012737  045010  001106            MOV     #TEST35,$LPADR  ;LOAD LOOP ON TEST ADDRESS
044774  012737  045010  001110            MOV     #TEST35,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
045002  012737  000004  001170            MOV     #4,$TIMES       ;;DO 4 ITERATIONS

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

045010                          TEST35:

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

045010  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
045016  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE
045022  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
045030  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
045036  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
045044  005060  000012                    CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
045050  012760  000011  000000            MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
045056  012760  000013  000000            MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
```

```
7330                                    ;;**************************************************************************
7331
7332                                            ;SEIZE THE DRIVE THROUGH PORT A
7333
7334  045064  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
7335  045072  013737  001216  001230            MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
7336  045100  005060  000012                    CLR     RHDS1(R0)       ;WRITE RHDS1
7337  045104  013737  001220  001232            MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
7338
7339                                    ;;**************************************************************************
7340                                    ;WAIT 500 MS
7341
7342
7343                                    ;;**************************************************************************
7344                                    ;START THE TIMER
7345
7346  045112  005037  001244                    CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7347  045116  012737  000764  001246            MOV     #500.,WATCH     ;SET WATCH TO 500 MS
7348  045124  005737  001246          1$:       TST     WATCH           ;WATCH EQUAL TO ZERO
7349  045130  001375                            BNE     1$              ;BR IF NOT
7350
7351                                    ;;**************************************************************************
7352                                    ;START THE TIMER
7353
7354  045132  005037  001244                    CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7355  045136  012737  003720  001246            MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
7356
7357                                    ;;**************************************************************************
7358                                    ;RETRIGGER THE TIMEOUT ONE-SHOT
7359
7360  045144  005760  000012                    TST     RHDS1(R0)       ;RETRIGGER THE ONE-SHOT
7361  045150  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
7362  045156  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7363  045164  005760  000012          2$:       TST     RHDS1(R0)       ;WAIT FOR TIMEOUT
7364  045170  001004                            BNE     3$              ;BR IF TIMEOUT OCCURED
7365  045172  005737  001246                    TST     WATCH           ;WATCH EQUAL TO ZERO ?
7366  045176  001372                            BNE     2$              ;BR IF NOT
7367  045200  104036                            ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7368  045202  013737  001244  001264  3$:       MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7369
7370                                    ;;**************************************************************************
7371
7372                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7373
7374  045210  005037  001242                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7375  045214  012737  000012  001122            MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
7376  045222  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7377  045226  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7378  045234  113760  001216  000010            MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7379  045242  016037  000012  001162            MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7380  045250  013737  001162  001156            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7381  045256  042737  100100  001156            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7382  045264  113760  001220  000010            MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7383  045272  016037  000012  001164            MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7384  045300  013737  001164  001160            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7385  045306  042737  100100  001160            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```

```
7386  045314  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7387  045322  001006                          BNE     64$             ;BR IF NOT
7388  045324  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7389  045330  001045                          BNE     66$             ;BR IF NOT
7390  045332  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7391  045334  000137  045520                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
7392  045340  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7393  045346  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7394  045354  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7395  045362  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7396  045366  001414                          BEQ     65$             ;BR IF ZERO
7397  045370  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7398  045376  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7399  045404  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7400  045412  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7401  045416  001012                          BNE     66$             ;BR IF NOT
7402  045420  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7403  045426  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
7404  045434  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7405  045442  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
7406  045444  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
7407  045452  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7408  045460  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7409  045466  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
7410  045470  104007                          ERROR   7               ;REPORT ERROR
7411  045472  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
7412  045500  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7413  045506  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7414  045514  001401                          BEQ     68$             ;BR IF OK
7415  045516  104007                          ERROR   7               ;REPORT ERROR
7416  045520  000240                  68$:    NOP
7417
7418                                          ;;*****************************************************************
7419                                          ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
7420
7421  045522  023737  001264  001252          CMP     TIMES,TIMEAP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7422  045530  003004                          BGT     4$              ;BR IF GREATER
7423  045532  023737  001264  001254          CMP     TIMES,TIMEAM    ;MEASURED TIME LESS THAN -25% TOLERANCE
7424  045540  002001                          BGE     .+4             ;BR IF NOT
7425  045542  104025                  4$:     ERROR   25              ;REPORT THE ERROR
7426
7427                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
7428
7429  045544  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
7430  045550  001412                          BEQ     TST36           ;;BR IF NOT
7431  045552  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
7432  045560  001406                          BEQ     TST36           ;;BR IF NOT
7433  045562  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
7434  045566  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
7435  045572  000177  133312                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
7436
7437                                          ;*************************************************************
7438                                          ;*TEST 36       TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
7439                                          ;*
7440                                          ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
7441                                          ;*
```

```
                                        ;*  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 + OR - 25% OF THE SPECIFIED
                                        ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
                                        ;*
                                        ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
                                        ;*      BIT IS SET.
                                        ;*
                                        ;*****************************************************************
045576                                  TST36:
045576  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
045600  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
045604  001406                          BEQ     2$                      ;BR IF NOT
045606  100002                          BPL     1$                      ;BR IF JUST ENTERED TEST
045610  000137  002410                  JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
045614  012737  177777  001266  1$:     MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
045622  112737  000036  001102  2$:     MOVB    #36,$TSTNM              ;TEST NUMBER
045630  012737  045652  001106          MOV     #TEST36,$LPADR          ;LOAD LOOP ON TEST ADDRESS
045636  012737  045652  001110          MOV     #TEST36,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
045644  012737  000004  001170          MOV     #4,$TIMES               ;;DO 4 ITERATIONS

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

045652                                  TEST36:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

045652  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
045660  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
045664  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
045672  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
045700  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
045706  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
045712  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
045720  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;;*************************************************************************

                                                ;SEIZE THE DRIVE THROUGH PORT B

045726  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
045734  013737  001220  001230                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
045742  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
045746  013737  001216  001232                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;;*************************************************************************
                                        ;WAIT 500 MS


                                        ;;*************************************************************************
                                        ;START THE TIMER
```

```
7498  045754  005037  001244                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7499  045760  012737  000764  001246          MOV     #500.,WATCH     ;SET WATCH TO 500 MS
7500  045766  005737  001246          1$:     TST     WATCH           ;WATCH EQUAL TO ZERO
7501  045772  001375                  BNE     1$              ;BR IF NOT
7502
7503                                  ;;**************************************************************************
7504                                  ;START THE TIMER
7505
7506  045774  005037  001244                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7507  046000  012737  003720  001246          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
7508
7509                                  ;;**************************************************************************
7510                                  ;RETRIGGER THE TIMEOUT ONE-SHOT
7511
7512  046006  005760  000012                  TST     RHDS1(R0)       ;RETRIGGER THE ONE-SHOT
7513  046012  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
7514  046020  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7515  046026  005760  000012          2$:     TST     RHDS1(R0)       ;WAIT FOR TIMEOUT
7516  046032  001004                  BNE     3$              ;BR IF TIMEOUT OCCURED
7517  046034  005737  001246                  TST     WATCH           ;WATCH EQUAL TO ZERO ?
7518  046040  001372                  BNE     2$              ;BR IF NOT
7519  046042  104036                  ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7520  046044  013737  001244  001264  3$:     MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7521
7522                                  ;;**************************************************************************
7523
7524                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7525
7526  046052  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7527  046056  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
7528  046064  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7529  046070  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7530  046076  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7531  046104  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7532  046112  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7533  046120  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7534  046126  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7535  046134  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7536  046142  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7537  046150  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7538  046156  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7539  046164  001006                  BNE     64$             ;BR IF NOT
7540  046166  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7541  046172  001045                  BNE     66$             ;BR IF NOT
7542  046174  104046                  ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7543  046176  000137  046362                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
7544  046202  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7545  046210  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7546  046216  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7547  046224  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7548  046230  001414                  BEQ     65$             ;BR IF ZERO
7549  046232  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7550  046240  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7551  046246  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7552  046254  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7553  046260  001012                  BNE     66$             ;BR IF NOT
```

```
7554  046262  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7555  046270  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
7556  046276  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7557  046304  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
7558  046306  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
7559  046314  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7560  046322  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7561  046330  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
7562  046332  104007                          ERROR   7               ;REPORT ERROR
7563  046334  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
7564  046342  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7565  046350  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7566  046356  001401                          BEQ     68$             ;BR IF OK
7567  046360  104007                          ERROR   7               ;REPORT ERROR
7568  046362  000240                  68$:    NOP
7569
7570                                  ;;*****************************************************************************
7571                                  ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
7572
7573  046364  023737  001264  001260          CMP     TIMES,TIMEBP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7574  046372  003004                          BGT     4$              ;BR IF GREATER
7575  046374  023737  001264  001262          CMP     TIMES,TIMEBM    ;MEASURED TIME LESS THAN -25% TOLERANCE
7576  046402  002001                          BGE     .+4             ;BR IF NOT
7577  046404  104025                  4$:     ERROR   25              ;REPORT THE ERROR
7578
7579                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
7580
7581  046406  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
7582  046412  001412                          BEQ     TST37           ;;BR IF NOT
7583  046414  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
7584  046422  001406                          BEQ     TST37           ;;BR IF NOT
7585  046424  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
7586  046430  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
7587  046434  000177  132450                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
7588
7589
7590                                  ;*****************************************************************
7591                                  ;*TEST 37       TEST NO TIMEOUT THROUGH PORT 'A'
7592                                  ;*
7593                                  ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
7594                                  ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
7595                                  ;*
7596                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
7597                                  ;*
7598                                  ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RHDS1 FROM PORT 'A'.
7599                                  ;*
7600                                  ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B'.  VERIFY THAT THE DRIVE
7601                                  ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
7602                                  ;*      SET FOR PORT 'B'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
7603                                  ;*
7604                                  ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
7605                                  ;*      BEEN RELEASED.
7606                                  ;*
7607                                  ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
7608                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7609                                  ;*
```

```
                                        ;*****************************************************************
046440                                  TST37:
046440  000004                                SCOPE                   ;INITIALIZE THE SCOPE HANDLER
046442  005737  001266                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
046446  001406                              BEQ     2$              ;BR IF NOT
046450  100002                              BPL     1$              ;BR IF JUST ENTERED TEST
046452  000137  002410                      JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
046456  012737  177777  001266  1$:        MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
046464  112737  000037  001102  2$:        MOVB    #37,$TSTNM      ;TEST NUMBER
046472  012737  046514  001106              MOV     #TEST37,$LPADR  ;LOAD LOOP ON TEST ADDRESS
046500  012737  046514  001110              MOV     #TEST37,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
046506  012737  000004  001170              MOV     #4,$TIMES       ;;DO 4 ITERATIONS

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

046514                                  TEST37:

                                            ;CLEAR ATTENTION BITS FOR BOTH PORTS

046514  113760  001216  000010              MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
046522  005060  000012                      CLR     RHDS1(R0)       ;SEIZE THE DRIVE
046526  012760  000011  000000              MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
046534  012760  000013  000000              MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
046542  113760  001220  000010              MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
046550  005060  000012                      CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
046554  012760  000011  000000              MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
046562  012760  000013  000000              MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;;*************************************************************************

                                            ;SEIZE THE DRIVE THROUGH PORT B

046570  113760  001220  000010              MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
046576  013737  001220  001230              MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
046604  005060  000012                      CLR     RHDS1(R0)       ;WRITE RHDS1
046610  013737  001216  001232              MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
046616  113760  001216  000010              MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
046624  013737  001216  001226              MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;*************************************************************************
                                        ;SET REQUEST THROUGH PORT A

046632  005060  000012                      CLR     RHDS1(R0)       ;SET REQUEST FOR PORT A
046636  113760  001220  000010              MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
046644  013737  001220  001226              MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;*************************************************************************
                                        ;RELEASE THE DRIVE THR

046652  012760  000013  000000              MOV     #13,RHCS1(R0)   ;RELEASE DRIVE THROUGH PORT B

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

```
7666  046660  013737  001252  001246          MOV     TIMEAP,WATCH    ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
7667
7668                                          ;;*************************************************************************
7669                                          ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
7670
7671  046666  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7672  046672  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
7673  046700  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7674  046706  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7675  046712  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7676  046716  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7677  046724  001403                          BEQ     64$             ;BR IF OK
7678  046726  104031                          ERROR   31              ;TYPE MESSAGE 31
7679  046730  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7680  046734  000240                  64$:    NOP
7681  046736  005737  001236                  TST     CKERR           ;REGISTER OK ?
7682  046742  001402                          BEQ     .+6             ;BR IF OK
7683  046744  000137  047402                  JMP     1$              ;BYPASS REST OF TEST IF NOT
7684  046750  005737  001246                  TST     WATCH           ;WATCH EQUAL ZERO ?
7685  046754  001375                          BNE     .-4             ;BR IF NOT
7686
7687                                          ;;*************************************************************************
7688                                          ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
7689
7690  046756  013737  001216  001226          MOV     PORTA,PTNBR     ;PORT NUMBER FOR TYPEOUT
7691  046764  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7692  046770  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
7693  046776  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7694  047004  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7695  047010  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7696  047014  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7697  047022  001403                          BEQ     65$             ;BR IF OK
7698  047024  104035                          ERROR   35              ;TYPE MESSAGE 35
7699  047026  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7700  047032  000240                  65$:    NOP
7701  047034  005737  001236                  TST     CKERR           ;REGISTER OK ?
7702  047040  001402                          BEQ     .+6             ;BR IF OK
7703  047042  000137  047402                  JMP     1$              ;BYPASS REST OF TEST IF NOT
7704
7705                                          ;;*************************************************************************
7706
7707                                          ;RELEASE THE DRIVE FROM PORT A
7708
7709  047046  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
7710  047054  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7711  047062  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7712
7713                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7714
7715  047070  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7716  047074  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
7717  047102  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7718  047106  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7719  047114  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7720  047122  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7721  047130  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
7722  047136  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7723  047144  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7724  047152  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7725  047160  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7726  047166  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7727  047174  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7728  047202  001006                          BNE     66$             ;BR IF NOT
7729  047204  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7730  047210  001045                          BNE     68$             ;BR IF NOT
7731  047212  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7732  047214  000137  047400                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
7733  047220  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7734  047226  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7735  047234  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7736  047242  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7737  047246  001414                          BEQ     67$             ;BR IF ZERO
7738  047250  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7739  047256  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7740  047264  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7741  047272  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7742  047276  001012                          BNE     68$             ;BR IF NOT
7743  047300  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7744  047306  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
7745  047314  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7746  047322  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7747  047324  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
7748  047332  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7749  047340  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7750  047346  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
7751  047350  104007                          ERROR   7               ;REPORT ERROR
7752  047352  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
7753  047360  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7754  047366  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7755  047374  001401                          BEQ     70$             ;BR IF OK
7756  047376  104007                          ERROR   7               ;REPORT ERROR
7757  047400  000240                  70$:    NOP
7758
7759  047402                          1$:
7760
7761                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
7762
7763  047402  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
7764  047406  001412                          BEQ     TST40           ;;BR IF NOT
7765  047410  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
7766  047416  001406                          BEQ     TST40           ;;BR IF NOT
7767  047420  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
7768  047424  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
7769  047430  000177  131454                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
7770
7771                                  ;*****************************************************************
7772                                  ;*TEST 40       TEST NO TIMEOUT THROUGH PORT 'B'
7773                                  ;*
7774                                  ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
7775                                  ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
7776                                  ;*
7777                                  ;*  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.
                                        ;*
                                        ;*****************************************************************
047434                                  TST40:
047434  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
047436  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
047442  001406                                  BEQ     2$              ;BR IF NOT
047444  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
047446  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
047452  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
047460  112737  000040  001102  2$:     MOVB    #40,$TSTNM      ;TEST NUMBER
047466  012737  047510  001106          MOV     #TEST40,$LPADR  ;LOAD LOOP ON TEST ADDRESS
047474  012737  047510  001110          MOV     #TEST40,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
047502  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS

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

047510                                  TEST40:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

047510  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
047516  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
047522  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
047530  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
047536  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
047544  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
047550  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
047556  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;;*************************************************************************

                                                ;SEIZE THE DRIVE THROUGH PORT A

047564  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
047572  013737  001216  001230                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
047600  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
047604  013737  001220  001232                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
047612  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
047620  013737  001220  001226                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;*************************************************************************
                                        ;SET REQUEST THROUGH PORT B
```

```
7834
7835  047626  005060  000012                  CLR     RHDS1(R0)          ;SET REQUEST FOR PORT B
7836  047632  113760  001216  000010          MOVB    PORTA,RHCS2(R0)    ;SELECT PORT A
7837  047640  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7838
7839                                  ;;**********************************************************************
7840                                  ;RELEASE THE DRIVE THR
7841
7842  047646  012760  000013  000000          MOV     #13,RHCS1(R0)      ;RELEASE DRIVE THROUGH PORT A
7843
7844                                  ;;**********************************************************************
7845                                  ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
7846
7847  047654  013737  001260  001246          MOV     TIMEBP,WATCH       ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
7848
7849                                  ;;**********************************************************************
7850                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
7851
7852  047662  005037  001236                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
7853  047666  016037  000012  001126          MOV     RHDS1(R0),$BDDAT   ;GET CONTENTS OF RHDS1
7854  047674  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7855  047702  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
7856  047706  005037  001124                  CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
7857  047712  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
7858  047720  001403                          BEQ     64$                ;BR IF OK
7859  047722  104031                          ERROR   31                 ;TYPE MESSAGE 31
7860  047724  005137  001236                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
7861  047730  000240                  64$:    NOP
7862  047732  005737  001236                  TST     CKERR              ;REGISTER OK ?
7863  047736  001402                          BEQ     .+6                ;BR IF OK
7864  047740  000137  050376                  JMP     1$                 ;BYPASS REST OF TEST IF NOT
7865  047744  005737  001246                  TST     WATCH              ;WATCH EQUAL ZERO ?
7866  047750  001375                          BNE     .-4                ;BR IF NOT
7867
7868                                  ;;**********************************************************************
7869                                  ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
7870
7871  047752  013737  001220  001226          MOV     PORTB,PTNBR        ;PORT NUMBER FOR TYPEOUT
7872  047760  005037  001236                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
7873  047764  016037  000012  001126          MOV     RHDS1(R0),$BDDAT   ;GET CONTENTS OF RHDS1
7874  047772  012737  000012  001122          MOV     #RHDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7875  050000  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
7876  050004  005037  001124                  CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
7877  050010  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
7878  050016  001403                          BEQ     65$                ;BR IF OK
7879  050020  104035                          ERROR   35                 ;TYPE MESSAGE 35
7880  050022  005137  001236                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
7881  050026  000240                  65$:    NOP
7882  050030  005737  001236                  TST     CKERR              ;REGISTER OK ?
7883  050034  001402                          BEQ     .+6                ;BR IF OK
7884  050036  000137  050376                  JMP     1$                 ;BYPASS REST OF TEST IF NOT
7885
7886                                  ;;**********************************************************************
7887
7888                                          ;RELEASE THE DRIVE FROM PORT B
7889
```

```
7890  050042  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
7891  050050  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7892  050056  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
7893
7894                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7895
7896  050064  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7897  050070  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
7898  050076  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7899  050102  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7900  050110  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7901  050116  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7902  050124  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7903  050132  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7904  050140  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7905  050146  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7906  050154  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7907  050162  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7908  050170  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7909  050176  001006                          BNE     66$             ;BR IF NOT
7910  050200  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7911  050204  001045                          BNE     68$             ;BR IF NOT
7912  050206  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7913  050210  000137  050374                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
7914  050214  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7915  050222  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7916  050230  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
7917  050236  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7918  050242  001414                          BEQ     67$             ;BR IF ZERO
7919  050244  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7920  050252  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7921  050260  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
7922  050266  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7923  050272  001012                          BNE     68$             ;BR IF NOT
7924  050274  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7925  050302  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
7926  050310  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7927  050316  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7928  050320  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
7929  050326  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7930  050334  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7931  050342  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
7932  050344  104007                          ERROR   7               ;REPORT ERROR
7933  050346  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
7934  050354  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7935  050362  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7936  050370  001401                          BEQ     70$             ;BR IF OK
7937  050372  104007                          ERROR   7               ;REPORT ERROR
7938  050374  000240                  70$:    NOP
7939
7940  050376                          1$:
7941
7942                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
7943
7944  050376  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
7945  050402  001412                          BEQ     TST41           ;;BR IF NOT
```

```
7946  050404  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
7947  050412  001406                          BEQ     TST41           ;;BR IF NOT
7948  050414  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
7949  050420  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
7950  050424  000177  130460                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
7951
7952
7953                                  ;*****************************************************************
7954                                  ;*TEST 41        TEST PORT 'A' ATTENTION AFTER A COMMAND
7955                                  ;*
7956                                  ;*
7957                                  ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
7958                                  ;*      COMMAND.
7959                                  ;*
7960                                  ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
7961                                  ;*
7962                                  ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
7963                                  ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
7964                                  ;*      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
7965                                  ;*
7966                                  ;*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
7967                                  ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7968                                  ;*
7969                                  ;*****************************************************************
7970  050430                          TST41:
7971  050430  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
7972  050432  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7973  050436  001406                          BEQ     2$              ;BR IF NOT
7974  050440  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
7975  050442  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
7976  050446  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
7977  050454  112737  000041  001102  2$:     MOVB    #41,$TSTNM      ;TEST NUMBER
7978  050462  012737  050504  001106          MOV     #TST41,$LPADR   ;LOAD LOOP ON TEST ADDRESS
7979  050470  012737  050504  001110          MOV     #TST41,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
7980  050476  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
7981
7982
7983                                  ;;*****************************************************************
7984                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
7985
7986  050504                          TEST41:
7987
7988                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7989
7990  050504  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
7991  050512  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
7992  050516  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
7993  050524  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7994  050532  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
7995  050540  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7996  050544  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
7997  050552  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
7998  050560  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
7999  050566  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8000  050574  013737  001216  001230          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
8001
```

```
8002                                    ;;*************************************************************************
8003                                    ;DO A RECALIBRATE THROUGH PORT A
8004
8005  050602  012760  000007  000000            MOV     #7,RHCS1(R0)    ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
8006
8007                                    ;;*************************************************************************
8008                                    ;WAIT FOR DRIVE TO FINISH
8009
8010  050610  032760  000200  000012            BIT     #DRY,RHDS1(R0)  ;WAIT FOR DRIVE TO FINISH
8011  050616  001774                            BEQ     .-6             ;BR IF NOT FINISHED
8012
8013                                    ;;*************************************************************************
8014                                    ;CONFIRM THAT ATTENTION IS SET FOR PORT A
8015
8016  050620  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8017  050624  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
8018  050632  012737  000012  001122            MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8019  050640  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8020  050644  012737  100000  001124            MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
8021  050652  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
8022  050660  042737  077777  001156            BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
8023  050666  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
8024  050674  001414                            BEQ     64$             ;BR IF OK
8025  050676  013737  001126  001166            MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
8026  050704  042737  100000  001166            BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
8027  050712  053737  001166  001124            BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
8028  050720  104032                            ERROR   32              ;TYPE MESSAGE 32
8029  050722  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8030  050726  000240                    64$:    NOP
8031  050730  113760  001220  000010            MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8032  050736  013737  001220  001226            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8033
8034                                    ;;*************************************************************************
8035                                    ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
8036
8037  050744  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8038  050750  016037  000012  001126            MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
8039  050756  012737  000012  001122            MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8040  050764  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8041  050770  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8042  050774  013737  001126  001156            MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
8043  051002  042737  077777  001156            BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
8044  051010  023737  001124  001156            CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
8045  051016  001414                            BEQ     65$             ;BR IF OK
8046  051020  013737  001126  001166            MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
8047  051026  042737  100000  001166            BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
8048  051034  053737  001166  001124            BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
8049  051042  104032                            ERROR   32              ;TYPE MESSAGE 32
8050  051044  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8051  051050  000240                    65$:    NOP
8052
8053                                    ;;*************************************************************************
8054
8055                                            ;RELEASE THE DRIVE FROM PORT A
8056
8057  051052  113760  001216  000010            MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
```

```
8058  051060  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8059  051066  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8060
8061                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8062
8063  051074  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8064  051100  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
8065  051106  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8066  051112  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8067  051120  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8068  051126  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8069  051134  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8070  051142  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8071  051150  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8072  051156  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8073  051164  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8074  051172  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8075  051200  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8076  051206  001006                          BNE     66$             ;BR IF NOT
8077  051210  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8078  051214  001045                          BNE     68$             ;BR IF NOT
8079  051216  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8080  051220  000137  051404                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
8081  051224  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8082  051232  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8083  051240  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8084  051246  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8085  051252  001414                          BEQ     67$             ;BR IF ZERO
8086  051254  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8087  051262  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8088  051270  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8089  051276  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8090  051302  001012                          BNE     68$             ;BR IF NOT
8091  051304  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8092  051312  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
8093  051320  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8094  051326  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
8095  051330  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
8096  051336  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8097  051344  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
8098  051352  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
8099  051354  104007                          ERROR   7               ;REPORT ERROR
8100  051356  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
8101  051364  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8102  051372  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
8103  051400  001401                          BEQ     70$             ;BR IF OK
8104  051402  104007                          ERROR   7               ;REPORT ERROR
8105  051404  000240                  70$:    NOP
8106
8107                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
8108
8109  051406  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
8110  051412  001412                          BEQ     TST42           ;;BR IF NOT
8111  051414  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
8112  051422  001406                          BEQ     TST42           ;;BR IF NOT
8113  051424  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
```

```
8114  051430  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
8115  051434  000177  127450                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
8116
8117                                  ;*****************************************************************
8118                                  ;*TEST 42       TEST PORT 'B' ATTENTION AFTER A COMMAND
8119                                  ;*
8120                                  ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
8121                                  ;*      COMMAND.
8122                                  ;*
8123                                  ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
8124                                  ;*
8125                                  ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
8126                                  ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND
8127                                  ;*      THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
8128                                  ;*
8129                                  ;*  C.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
8130                                  ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
8131                                  ;*
8132                                  ;*****************************************************************
8133  051440                          TST42:
8134  051440  000004                         SCOPE                   ;INITIALIZE THE SCOPE HANDLER
8135  051442  005737  001266                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
8136  051446  001406                         BEQ     2$              ;BR IF NOT
8137  051450  100002                         BPL     1$              ;BR IF JUST ENTERED TEST
8138  051452  000137  002410                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
8139  051456  012737  177777  001266  1$:    MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
8140  051464  112737  000042  001102  2$:    MOVB    #42,$TSTNM      ;TEST NUMBER
8141  051472  012737  051514  001106          MOV     #TEST42,$LPADR  ;LOAD LOOP ON TEST ADDRESS
8142  051500  012737  051514  001110          MOV     #TEST42,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
8143  051506  012737  000004  001170          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
8144
8145
8146                                  ;;*****************************************************************
8147                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
8148
8149  051514                          TEST42:
8150
8151                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
8152
8153  051514  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
8154  051522  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
8155  051526  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
8156  051534  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8157  051542  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
8158  051550  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
8159  051554  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
8160  051562  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8161  051570  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
8162  051576  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8163  051604  013737  001220  001230          MOV     PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS
8164
8165                                  ;;*****************************************************************
8166                                  ;DO A RECALIBRATE THROUGH PORT B
8167
8168  051612  012760  000007  000000          MOV     #7,RHCS1(R0)    ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B
8169
```

```
8170                                    ;;*************************************************************************
8171                                    ;WAIT FOR DRIVE TO FINISH
8172
8173  051620  032760  000200  000012          BIT     #DRY,RHDS1(R0)  ;WAIT FOR DRIVE TO FINISH
8174  051626  001774                          BEQ     .-6             ;BR IF NOT FINISHED
8175
8176                                    ;;*************************************************************************
8177                                    ;CONFIRM THAT ATTENTION IS SET FOR PORT B
8178
8179  051630  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8180  051634  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
8181  051642  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8182  051650  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8183  051654  012737  100000  001124          MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
8184  051662  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
8185  051670  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
8186  051676  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
8187  051704  001414                          BEQ     64$             ;BR IF OK
8188  051706  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
8189  051714  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
8190  051722  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
8191  051730  104032                          ERROR   32              ;TYPE MESSAGE 32
8192  051732  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8193  051736  000240                  64$:    NOP
8194  051740  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8195  051746  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8196
8197                                    ;;*************************************************************************
8198                                    ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A
8199
8200  051754  005037  001236                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8201  051760  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;GET CONTENTS OF RHDS1
8202  051766  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8203  051774  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8204  052000  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8205  052004  013737  001126  001156          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
8206  052012  042737  077777  001156          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
8207  052020  023737  001124  001156          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
8208  052026  001414                          BEQ     65$             ;BR IF OK
8209  052030  013737  001126  001166          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
8210  052036  042737  100000  001166          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
8211  052044  053737  001166  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
8212  052052  104032                          ERROR   32              ;TYPE MESSAGE 32
8213  052054  005137  001236                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8214  052060  000240                  65$:    NOP
8215
8216                                    ;;*************************************************************************
8217
8218                                    ;RELEASE THE DRIVE FROM PORT B
8219
8220  052062  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8221  052070  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8222  052076  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8223
8224                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8225
```

```
8226  052104  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8227  052110  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
8228  052116  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8229  052122  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8230  052130  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8231  052136  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8232  052144  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8233  052152  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8234  052160  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8235  052166  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8236  052174  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8237  052202  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8238  052210  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8239  052216  001006                          BNE     66$             ;BR IF NOT
8240  052220  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8241  052224  001045                          BNE     68$             ;BR IF NOT
8242  052226  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8243  052230  000137  052414                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
8244  052234  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8245  052242  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8246  052250  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8247  052256  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8248  052262  001414                          BEQ     67$             ;BR IF ZERO
8249  052264  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8250  052272  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8251  052300  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8252  052306  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8253  052312  001012                          BNE     68$             ;BR IF NOT
8254  052314  012737  177777  001242  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8255  052322  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
8256  052330  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8257  052336  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
8258  052340  013737  001162  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
8259  052346  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8260  052354  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
8261  052362  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
8262  052364  104007                          ERROR   7               ;REPORT ERROR
8263  052366  013737  001164  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
8264  052374  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8265  052402  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
8266  052410  001401                          BEQ     70$             ;BR IF OK
8267  052412  104007                          ERROR   7               ;REPORT ERROR
8268  052414  000240                  70$:    NOP
8269
8270                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
8271
8272  052416  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
8273  052422  001412                          BEQ     TST43           ;;BR IF NOT
8274  052424  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
8275  052432  001406                          BEQ     TST43           ;;BR IF NOT
8276  052434  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
8277  052440  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
8278  052444  000177  126440                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
8279
8280
8281                                  ;*****************************************************************
```

```
                                ;*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.
                                ;*
                                ;*****************************************************************
052450                          TST43:
052450  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
052452  005737  001266                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
052456  001406                          BEQ     2$              ;BR IF NOT
052460  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
052462  000137  002410                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
052466  012737  177777  001266 1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
052474  112737  000043  001102 2$:     MOVB    #43,$TSTNM      ;TEST NUMBER
052502  012737  052524  001106         MOV     #TEST43,$LPADR  ;LOAD LOOP ON TEST ADDRESS
052510  012737  052524  001110         MOV     #TEST43,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
052516  012737  007640  001170         MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS

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

052524                          TEST43:

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

052524  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
052532  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
052536  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
052544  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
052552  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
052560  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
052564  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
052572  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

052600  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
```

```
8338  052606  013737  001216  001230          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
8339  052614  005060  000012                  CLR     RHDS1(R0)     ;WRITE RHDS1
8340  052620  013737  001220  001232          MOV     PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
8341  052626  012760  177777  000014          MOV     #-1,RHER1(R0) ;LOAD 1'S INTO RHER1 THROUGH PORT A
8342  052634  012760  177777  000040          MOV     #-1,RHER2(R0) ;LOAD 1'S INTO RHER2 THROUGH PORT A
8343  052642  012760  177777  000042          MOV     #-1,RHER3(R0) ;LOAD 1'S INTO RHER3 THROUGH PORT A
8344  052650  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
8345  052656  013737  001220  001226          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8346  052664  004737  053550                  JSR     PC,TST43B     ;CHECK THE REGISTERS THROUGH PORT B
8347  052670  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
8348  052676  013737  001216  001226          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8349  052704  005060  000042                  CLR     RHER3(R0)     ;CLEAR RHER3 ON PORT A
8350  052710  005060  000040                  CLR     RHER2(R0)     ;CLEAR RHER2 ON PORT A
8351  052714  005060  000014                  CLR     RHER1(R0)     ;CLEAR RHER1 ON PORT A
8352  052720  013760  001224  000016          MOV     ASR1,RHAS(R0) ;CLEAR THE ATTENTION BIT FOR PORT A
8353  052726  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
8354  052734  013737  001220  001226          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8355  052742  012760  177777  000014          MOV     #-1,RHER1(R0) ;LOAD 1'S INTO RHER1 THROUGH PORT B
8356  052750  012760  177777  000040          MOV     #-1,RHER2(R0) ;LOAD 1'S INTO RHER2 THROUGH PORT B
8357  052756  012760  177777  000042          MOV     #-1,RHER3(R0) ;LOAD 1'S INTO RHER3 THROUGH PORT B
8358  052764  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
8359  052772  013737  001216  001226          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8360  053000  004737  053550                  JSR     PC,TST43B     ;CHECK THE REGISTERS THROUGH PORT A
8361
8362                                          ;RELEASE THE DRIVE FROM PORT A
8363
8364  053004  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
8365  053012  013737  001216  001226          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8366  053020  012760  000013  000000          MOV     #13,RHCS1(R0) ;ISSUE RELEASE THROUGH PORT A
8367
8368                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
8369
8370  053026  005037  001242                  CLR     RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
8371  053032  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
8372  053040  012737  000012  001122          MOV     #RHDS1,$BDADR ;REGISTER ADDRESS INCREMENT
8373  053046  060037  001122                  ADD     R0,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
8374  053052  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
8375  053060  013737  001220  001226          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8376  053066  016037  000012  001156          MOV     RHDS1(R0),$TMPO ;READ STATUS REGISTER FROM PORT B
8377  053074  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
8378  053102  013737  001216  001226          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8379  053110  016037  000012  001126          MOV     RHDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT A
8380  053116  001404                          BEQ     64$           ;BR IF STATUS FROM PORT A ZERO
8381  053120  005737  001156                  TST     $TMPO         ;IS STATUS FROM PORT B ZERO ?
8382  053124  001401                          BEQ     64$           ;BR IF ZERO
8383  053126  104031                          ERROR   31            ;REPORT DRIVE IN NEUTRAL
8384  053130  013737  001156  001126  64$:   MOV     $TMPO,$BDDAT  ;CHECK STATUS FROM PORT B
8385  053136  013737  001220  001226          MOV     PORTB,PTNBR   ;CHANGE PORT ADDRESS FOR TYPEOUT
8386  053144  023737  001124  001126          CMP     $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
8387  053152  001401                          BEQ     .+4           ;BR IF OK
8388  053154  104027                          ERROR   27            ;REPORT REGISTER ERROR
8389
8390                                          ;RELEASE THE DRIVE FROM PORT B
8391
8392  053156  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
8393  053164  013737  001220  001226          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
8394  053172  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8395
8396                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8397
8398  053200  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8399  053204  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
8400  053212  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8401  053216  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8402  053224  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8403  053232  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8404  053240  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8405  053246  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8406  053254  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8407  053262  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8408  053270  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8409  053276  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8410  053304  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8411  053312  001006                          BNE     65$             ;BR IF NOT
8412  053314  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8413  053320  001045                          BNE     67$             ;BR IF NOT
8414  053322  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8415  053324  000137  053510                  JMP     69$             ;BYPASS THE REST OF THE CHECKS
8416  053330  013737  001162  001126  65$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8417  053336  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8418  053344  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8419  053352  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8420  053356  001414                          BEQ     66$             ;BR IF ZERO
8421  053360  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8422  053366  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8423  053374  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8424  053402  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8425  053406  001012                          BNE     67$             ;BR IF NOT
8426  053410  012737  177777  001242  66$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8427  053416  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
8428  053424  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8429  053432  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
8430  053434  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
8431  053442  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8432  053450  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
8433  053456  001401                          BEQ     68$             ;BR IF OK FROM PORT A.
8434  053460  104007                          ERROR   7               ;REPORT ERROR
8435  053462  013737  001164  001126  68$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
8436  053470  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8437  053476  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
8438  053504  001401                          BEQ     69$             ;BR IF OK
8439  053506  104007                          ERROR   7               ;REPORT ERROR
8440  053510  000240                  69$:    NOP
8441
8442                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
8443
8444  053512  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
8445  053516  001547                          BEQ     TST44           ;;BR IF NOT
8446  053520  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
8447  053526  001543                          BEQ     TST44           ;;BR IF NOT
8448  053530  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
8449  053534  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
```

```
 8450  053540  000177  125344                    JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
 8451  053544  000137  054036                    JMP     TST44           ;GO TO THE NEXT TEST
 8452
 8453                                            ;CHECK THE REGISTERS ON THE SELECTED PORT
 8454
 8455  053550                            TST43B:
 8456  053550  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 8457  053554  016037  000014  001126            MOV     RHER1(R0),$BDDAT  ;GET CONTENTS OF RHER1
 8458  053562  012737  000014  001122            MOV     #RHER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 8459  053570  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 8460  053574  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 8461  053600  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 8462  053606  001403                            BEQ     64$             ;BR IF OK
 8463  053610  104006                            ERROR   6               ;TYPE MESSAGE 6
 8464  053612  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 8465  053616  016037  000000  001126  64$:      MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 8466  053624  032737  020000  001126            BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 8467  053632  001404                            BEQ     .+12            ;BR IF NOT
 8468  053634  104011                            ERROR   11              ;REPORT THE ERROR
 8469  053636  012760  040000  000000            MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
 8470  053644  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 8471  053650  016037  000040  001126            MOV     RHER2(R0),$BDDAT  ;GET CONTENTS OF RHER2
 8472  053656  012737  000040  001122            MOV     #RHER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 8473  053664  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 8474  053670  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 8475  053674  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 8476  053702  001403                            BEQ     65$             ;BR IF OK
 8477  053704  104006                            ERROR   6               ;TYPE MESSAGE 6
 8478  053706  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 8479  053712  016037  000000  001126  65$:      MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 8480  053720  032737  020000  001126            BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 8481  053726  001404                            BEQ     .+12            ;BR IF NOT
 8482  053730  104011                            ERROR   11              ;REPORT THE ERROR
 8483  053732  012760  040000  000000            MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
 8484  053740  005037  001236                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 8485  053744  016037  000042  001126            MOV     RHER3(R0),$BDDAT  ;GET CONTENTS OF RHER3
 8486  053752  012737  000042  001122            MOV     #RHER3,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 8487  053760  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 8488  053764  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 8489  053770  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 8490  053776  001403                            BEQ     66$             ;BR IF OK
 8491  054000  104006                            ERROR   6               ;TYPE MESSAGE 6
 8492  054002  005137  001236                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 8493  054006  016037  000000  001126  66$:      MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 8494  054014  032737  020000  001126            BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 8495  054022  001404                            BEQ     .+12            ;BR IF NOT
 8496  054024  104011                            ERROR   11              ;REPORT THE ERROR
 8497  054026  012760  040000  000000            MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
 8498  054034  000207                            RTS     PC              ;RETURN
 8499
 8500                                    ;*****************************************************************
 8501                                    ;*TEST 44       TEST PORT INTERACTION FROM PORT 'B'
 8502                                    ;*
 8503                                    ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
 8504                                    ;*
 8505                                    ;*  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.
                                        ;*
                                        ;*************************************************************
054036                                  TST44:
054036  000004                                  SCOPE                   ;INITIALIZE THE SCOPE HANDLER
054040  005737  001266                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
054044  001406                                  BEQ     2$              ;BR IF NOT
054046  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
054050  000137  002410                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
054054  012737  177777  001266      1$:         MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
054062  112737  000044  001102      2$:         MOVB    #44,$TSTNM      ;TEST NUMBER
054070  012737  054112  001106                  MOV     #TEST44,$LPADR  ;LOAD LOOP ON TEST ADDRESS
054076  012737  054112  001110                  MOV     #TEST44,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
054104  012737  007640  001170                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS

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

054112                                  TEST44:

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

054112  113760  001216  000010                  MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
054120  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE
054124  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
054132  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
054140  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
054146  005060  000012                          CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
054152  012760  000011  000000                  MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
054160  012760  000013  000000                  MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE

                                                ;SEIZE THE DRIVE THROUGH PORT B.

054166  113760  001220  000010                  MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
054174  013737  001220  001230                  MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
054202  005060  000012                          CLR     RHDS1(R0)       ;WRITE RHDS1
054206  013737  001216  001232                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
054214  012760  177777  000014                  MOV     #-1,RHER1(R0)   ;LOAD 1'S INTO RHER1 THROUGH PORT B
054222  012760  177777  000040                  MOV     #-1,RHER2(R0)   ;LOAD 1'S INTO RHER2 THROUGH PORT B
```

```
8562  054230  012760  177777  000042          MOV     #-1,RHER3(R0)   ;LOAD 1'S INTO RHER3 THROUGH PORT B
8563  054236  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8564  054244  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8565  054252  004737  055136                  JSR     PC,TST44B       ;CHECK THE REGISTERS THROUGH PORT A
8566  054256  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8567  054264  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8568  054272  005060  000042                  CLR     RHER3(R0)       ;CLEAR RHER3 ON PORT B
8569  054276  005060  000040                  CLR     RHER2(R0)       ;CLEAR RHER2 ON PORT B
8570  054302  005060  000014                  CLR     RHER1(R0)       ;CLEAR RHER1 ON PORT B
8571  054306  013760  001224  000016          MOV     ASR1,RHAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT B
8572  054314  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8573  054322  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8574  054330  012760  177777  000014          MOV     #-1,RHER1(R0)   ;LOAD 1'S INTO RHER1 THROUGH PORT A
8575  054336  012760  177777  000040          MOV     #-1,RHER2(R0)   ;LOAD 1'S INTO RHER2 THROUGH PORT A
8576  054344  012760  177777  000042          MOV     #-1,RHER3(R0)   ;LOAD 1'S INTO RHER3 THROUGH PORT A
8577  054352  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8578  054360  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8579  054366  004737  055136                  JSR     PC,TST44B       ;CHECK THE REGISTERS THROUGH PORT B
8580
8581                                          ;RELEASE THE DRIVE FROM PORT B
8582
8583  054372  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8584  054400  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8585  054406  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8586
8587                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
8588
8589  054414  005037  001242                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
8590  054420  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8591  054426  012737  000012  001122          MOV     #RHDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
8592  054434  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
8593  054440  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8594  054446  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8595  054454  016037  000012  001156          MOV     RHDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
8596  054462  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8597  054470  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8598  054476  016037  000012  001126          MOV     RHDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
8599  054504  001404                          BEQ     64$             ;BR IF STATUS FROM PORT B ZERO
8600  054506  005737  001156                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
8601  054512  001401                          BEQ     64$             ;BR IF ZERO
8602  054514  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
8603  054516  013737  001156  001126  64$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
8604  054524  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
8605  054532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
8606  054540  001401                          BEQ     .+4             ;BR IF OK
8607  054542  104027                          ERROR   27              ;REPORT REGISTER ERROR
8608
8609                                          ;RELEASE THE DRIVE FROM PORT A
8610
8611  054544  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8612  054552  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8613  054560  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8614
8615                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8616
8617  054566  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
```

```
 8618  054572  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
 8619  054600  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 8620  054604  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 8621  054612  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
 8622  054620  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 8623  054626  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 8624  054634  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 8625  054642  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
 8626  054650  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 8627  054656  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 8628  054664  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 8629  054672  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 8630  054700  001006                          BNE     65$             ;BR IF NOT
 8631  054702  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 8632  054706  001045                          BNE     67$             ;BR IF NOT
 8633  054710  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 8634  054712  000137  055076                  JMP     69$             ;BYPASS THE REST OF THE CHECKS
 8635  054716  013737  001162  001126  65$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 8636  054724  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 8637  054732  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
 8638  054740  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 8639  054744  001414                          BEQ     66$             ;BR IF ZERO
 8640  054746  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 8641  054754  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 8642  054762  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
 8643  054770  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 8644  054774  001012                          BNE     67$             ;BR IF NOT
 8645  054776  012737  177777  001242  66$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 8646  055004  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
 8647  055012  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 8648  055020  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 8649  055022  013737  001162  001126  67$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
 8650  055030  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8651  055036  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8652  055044  001401                          BEQ     68$             ;BR IF OK FROM PORT A.
 8653  055046  104007                          ERROR   7               ;REPORT ERROR
 8654  055050  013737  001164  001126  68$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
 8655  055056  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8656  055064  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8657  055072  001401                          BEQ     69$             ;BR IF OK
 8658  055074  104007                          ERROR   7               ;REPORT ERROR
 8659  055076  000240                  69$:    NOP
 8660
 8661                                          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
 8662
 8663  055100  105737  001103                  TSTB    $ERFLG          ;DID AN ERROR OCCUR ?
 8664  055104  001547                          BEQ     TST45           ;;BR IF NOT
 8665  055106  032737  001000  177570          BIT     #SW09,SWR       ;SEE IF LOOP ON ERROR SET (SWR9=1)
 8666  055114  001543                          BEQ     TST45           ;;BR IF NOT
 8667  055116  105037  001103                  CLRB    $ERFLG          ;CLEAR THE ERROR FLAG
 8668  055122  005037  001170                  CLR     $TIMES          ;CLEAR THE MAX ITERATION COUNT
 8669  055126  000177  123756                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
 8670  055132  000137  055424                  JMP     TST45           ;GO TO THE NEXT TEST
 8671
 8672                                          ;CHECK THE REGISTERS ON THE SELECTED PORT
 8673
```

```
8674  055136                                  TST44B:
8675  055136  005037  001236                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8676  055142  016037  000014  001126                MOV     RHER1(R0),$BDDAT  ;GET CONTENTS OF RHER1
8677  055150  012737  000014  001122                MOV     #RHER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8678  055156  060037  001122                        ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8679  055162  005037  001124                        CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8680  055166  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8681  055174  001403                                BEQ     64$             ;BR IF OK
8682  055176  104006                                ERROR   6               ;TYPE MESSAGE 6
8683  055200  005137  001236                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8684  055204  016037  000000  001126  64$:          MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
8685  055212  032737  020000  001126                BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
8686  055220  001404                                BEQ     .+12            ;BR IF NOT
8687  055222  104011                                ERROR   11              ;REPORT THE ERROR
8688  055224  012760  040000  000000                MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
8689  055232  005037  001236                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8690  055236  016037  000040  001126                MOV     RHER2(R0),$BDDAT  ;GET CONTENTS OF RHER2
8691  055244  012737  000040  001122                MOV     #RHER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8692  055252  060037  001122                        ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8693  055256  005037  001124                        CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8694  055262  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8695  055270  001403                                BEQ     65$             ;BR IF OK
8696  055272  104006                                ERROR   6               ;TYPE MESSAGE 6
8697  055274  005137  001236                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8698  055300  016037  000000  001126  65$:          MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
8699  055306  032737  020000  001126                BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
8700  055314  001404                                BEQ     .+12            ;BR IF NOT
8701  055316  104011                                ERROR   11              ;REPORT THE ERROR
8702  055320  012760  040000  000000                MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
8703  055326  005037  001236                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8704  055332  016037  000042  001126                MOV     RHER3(R0),$BDDAT  ;GET CONTENTS OF RHER3
8705  055340  012737  000042  001122                MOV     #RHER3,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8706  055346  060037  001122                        ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8707  055352  005037  001124                        CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8708  055356  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8709  055364  001403                                BEQ     66$             ;BR IF OK
8710  055366  104006                                ERROR   6               ;TYPE MESSAGE 6
8711  055370  005137  001236                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8712  055374  016037  000000  001126  66$:          MOV     RHCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
8713  055402  032737  020000  001126                BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
8714  055410  001404                                BEQ     .+12            ;BR IF NOT
8715  055412  104011                                ERROR   11              ;REPORT THE ERROR
8716  055414  012760  040000  000000                MOV     #TRE,RHCS1(R0)  ;CLEAR 'MCPE'
8717  055422  000207                                RTS     PC              ;RETURN
8718
8719
8720                                  ;*************************************************************
8721                                  ;*TEST 45       TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
8722                                  ;*
8723                                  ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
8724                                  ;*
8725                                  ;*  A.  SET THE ATTENTION BIT FOR PORT 'A'.
8726                                  ;*
8727                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RHDS1.
8728                                  ;*
8729                                  ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
```

```
8730                                    ;*      FOR THE DRIVE IS SET.
8731                                    ;*
8732                                    ;*************************************************************
8733  055424                            TST45:
8734  055424  000004                           SCOPE                   ;INITIALIZE THE SCOPE HANDLER
8735  055426  005737  001266                   TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
8736  055432  001406                           BEQ     2$              ;BR IF NOT
8737  055434  100002                           BPL     1$              ;BR IF JUST ENTERED TEST
8738  055436  000137  002410                   JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
8739  055442  012737  177777  001266  1$:      MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
8740  055450  112737  000045  001102  2$:      MOVB    #45,$TSTNM      ;TEST NUMBER
8741  055456  012737  055500  001106           MOV     #TEST45,$LPADR  ;LOAD LOOP ON TEST ADDRESS
8742  055464  012737  055500  001110           MOV     #TEST45,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
8743  055472  012737  007640  001170           MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
8744
8745
8746                                    ;;*************************************************************************
8747                                    ;END OF 'SCOPE' SETUP - START OF MAIN TEST
8748
8749  055500                            TEST45:
8750
8751                                           ;CLEAR ATTENTION BITS FOR BOTH PORTS
8752
8753  055500  113760  001216  000010           MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
8754  055506  005060  000012                   CLR     RHDS1(R0)       ;SEIZE THE DRIVE
8755  055512  012760  000011  000000           MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
8756  055520  012760  000013  000000           MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8757  055526  113760  001220  000010           MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
8758  055534  005060  000012                   CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
8759  055540  012760  000011  000000           MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
8760  055546  012760  000013  000000           MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8761  055554  113760  001216  000010           MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8762  055562  012760  177777  000014           MOV     #-1,RHER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT A
8763  055570  005060  000014                   CLR     RHER1(R0)       ;CLEAR THE ERRORS
8764  055574  113760  001220  000010           MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8765  055602  005760  000012           1$:     TST     RHDS1(R0)       ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
8766  055606  001775                           BEQ     1$              ;BR IF STILL SEIZED BY PORT A
8767  055610  012737  000016  001122           MOV     #RHAS,$BDADR    ;FORM ADDRESS OF ATTN REG IF ERROR
8768  055616  060037  001122                   ADD     R0,$BDADR       ;ADD THE ADDRESS BASE
8769  055622  013737  001224  001124           MOV     ASR1,$GDDAT     ;GOOD DATA FOR ERROR MESSAGE
8770  055630  013737  001224  001160           MOV     ASR1,$TMP1      ;MAKE DATA COMPARE MASK
8771  055636  005137  001160                   COM     $TMP1           ;COMPLEMENT IT
8772  055642  012737  000102  001110           MOV     #'B,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
8773  055650  113760  001220  000010           MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8774  055656  013737  001220  001226           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8775  055664  013737  001220  001230           MOV     PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS
8776  055672  005060  000012                   CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT B
8777  055676  016037  000016  001126  2$:      MOV     RHAS(R0),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
8778  055704  013737  001126  001156           MOV     $BDDAT,$TMP0    ;PUT CONTENTS INTO WORKING LOCATION
8779  055712  043737  001160  001156           BIC     $TMP1,$TMP0     ;CLEAR OTHER BITS
8780  055720  023737  001124  001156           CMP     $GDDAT,$TMP0    ;SEE IF ATTN BIT FOR DRIVE SET
8781  055726  001411                           BEQ     3$              ;BR IF SET
8782  055730  104053                           ERROR   53              ;REPORT THE ERROR
8783  055732  005037  001170                   CLR     $TIMES          ;CLEAR ITERATION COUNT
8784  055736  032737  001000  177570           BIT     #SW09,SWR       ;LOOP ON THE ERROR ?
8785  055744  001402                           BEQ     .+6             ;BR IF SW09 NOT SET
```

```
8786  055746  000177  123136                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
8787  055752  005237  001104          3$:     INC     $ICNT           ;INCREMENT THE ITERATION COUNT
8788  055756  023737  001104  001170          CMP     $ICNT,$TIMES    ;DO THE SUBTEST SOME MORE ?
8789  055764  002002                          BGE     .+6             ;BR IF NOT
8790  055766  000137  055676                  JMP     2$              ;DO THE SUBTEST AGAIN
8791
8792                                          ;RELEASE THE DRIVE FROM PORT B
8793
8794  055772  113760  001220  000010          MOVB    PORTB,RHCS2(R0)  ;SELECT PORT B
8795  056000  013737  001220  001226          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8796  056006  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8797
8798                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8799
8800  056014  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8801  056020  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
8802  056026  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8803  056032  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
8804  056040  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8805  056046  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8806  056054  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8807  056062  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8808  056070  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8809  056076  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8810  056104  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8811  056112  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8812  056120  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8813  056126  001006                          BNE     64$             ;BR IF NOT
8814  056130  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8815  056134  001045                          BNE     66$             ;BR IF NOT
8816  056136  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8817  056140  000137  056340                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
8818  056144  013737  001162  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8819  056152  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8820  056160  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8821  056166  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8822  056172  001414                          BEQ     65$             ;BR IF ZERO
8823  056174  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8824  056202  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8825  056210  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8826  056216  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8827  056222  001012                          BNE     66$             ;BR IF NOT
8828  056224  012737  177777  001242  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8829  056232  012760  000011  000000          MOV     #11,RHCS1(R0)   ;CLEAR THE DRIVE
8830  056240  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
8831  056246  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
8832  056250  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
8833  056256  013737  001216  001226          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8834  056264  042737  100000  001162          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
8835  056272  023737  001124  001162          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
8836  056300  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
8837  056302  104007                          ERROR   7               ;REPORT ERROR
8838  056304  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
8839  056312  013737  001220  001226          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8840  056320  042737  100000  001164          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
8841  056326  023737  001124  001164          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
```

```
 8842  056334  001401                          BEQ     68$             ;BR IF OK
 8843  056336  104007                          ERROR   7               ;REPORT ERROR
 8844  056340  000240                  68$:    NOP
 8845
 8846                                  ;*****************************************************************
 8847                                  ;*TEST 46       TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
 8848                                  ;*
 8849                                  ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
 8850                                  ;*
 8851                                  ;*  A.  SET THE ATTENTION BIT FOR PORT 'B'.
 8852                                  ;*
 8853                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RHDS1.
 8854                                  ;*
 8855                                  ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
 8856                                  ;*      FOR THE DRIVE IS SET.
 8857                                  ;*
 8858                                  ;*****************************************************************
 8859  056342                          TST46:
 8860  056342  000004                          SCOPE                   ;INITIALIZE THE SCOPE HANDLER
 8861  056344  005737  001266                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 8862  056350  001406                          BEQ     2$              ;BR IF NOT
 8863  056352  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 8864  056354  000137  002410                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8865  056360  012737  177777  001266  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 8866  056366  112737  000046  001102  2$:     MOVB    #46,$TSTNM      ;TEST NUMBER
 8867  056374  012737  056416  001106          MOV     #TEST46,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8868  056402  012737  056416  001110          MOV     #TEST46,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 8869  056410  012737  007640  001170          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 8870
 8871
 8872                                  ;;*****************************************************************
 8873                                  ;END OF 'SCOPE' SETUP - START OF MAIN TEST
 8874
 8875  056416                          TEST46:
 8876
 8877                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 8878
 8879  056416  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT #A
 8880  056424  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE
 8881  056430  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
 8882  056436  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 8883  056444  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT #B
 8884  056452  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 8885  056456  012760  000011  000000          MOV     #11,RHCS1(R0)   ;ISSUE DRIVE CLEAR
 8886  056464  012760  000013  000000          MOV     #13,RHCS1(R0)   ;RELEASE THE DRIVE
 8887  056472  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B
 8888  056500  012760  177777  000014          MOV     #-1,RHER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT B
 8889  056506  005060  000014                  CLR     RHER1(R0)       ;CLEAR THE ERRORS
 8890  056512  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A
 8891  056520  005760  000012          1$:     TST     RHDS1(R0)       ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
 8892  056524  001775                          BEQ     1$              ;BR IF STILL SEIZED BY PORT B
 8893  056526  012737  000016  001122          MOV     #RHAS,$BDADR    ;FORM ADDRESS OF ATTN REG IF ERROR
 8894  056534  060037  001122                  ADD     R0,$BDADR       ;ADD THE ADDRESS BASE
 8895  056540  013737  001224  001124          MOV     ASR1,$GDDAT     ;GOOD DATA FOR ERROR MESSAGE
 8896  056546  013737  001224  001160          MOV     ASR1,$TMP1      ;MAKE DATA COMPARE MASK
 8897  056554  005137  001160                  COM     $TMP1           ;COMPLEMENT IT
```

```
8898  056560  012737  000102  001110          MOV     #'B,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
8899  056566  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8900  056574  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8901  056602  013737  001216  001230          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
8902  056610  005060  000012                  CLR     RHDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT A
8903  056614  016037  000016  001126  2$:     MOV     RHAS(R0),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
8904  056622  013737  001126  001156          MOV     $BDDAT,$TMP0    ;PUT CONTENTS INTO WORKING LOCATION
8905  056630  043737  001160  001156          BIC     $TMP1,$TMP0     ;CLEAR OTHER BITS
8906  056636  023737  001124  001156          CMP     $GDDAT,$TMP0    ;SEE IF ATTN BIT FOR DRIVE SET
8907  056644  001411                          BEQ     3$              ;BR IF SET
8908  056646  104053                          ERROR   53              ;REPORT THE ERROR
8909  056650  005037  001170                  CLR     $TIMES          ;CLEAR ITERATION COUNT
8910  056654  032737  001000  177570          BIT     #SW09,SWR       ;LOOP ON THE ERROR ?
8911  056662  001402                          BEQ     .+6             ;BR IF SW09 NOT SET
8912  056664  000177  122220                  JMP     @$LPERR         ;GO TO THE LOOP ADDRESS
8913  056670  005237  001104          3$:     INC     $ICNT           ;INCREMENT THE ITERATION COUNT
8914  056674  023737  001104  001170          CMP     $ICNT,$TIMES    ;DO THE SUBTEST SOME MORE ?
8915  056702  002002                          BGE     .+6             ;BR IF NOT
8916  056704  000137  056614                  JMP     2$              ;DO THE SUBTEST AGAIN
8917
8918                                          ;RELEASE THE DRIVE FROM PORT A
8919
8920  056710  113760  001216  000010          MOVB    PORTA,RHCS2(R0)  ;SELECT PORT A
8921  056716  013737  001216  001226          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8922  056724  012760  000013  000000          MOV     #13,RHCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8923
8924                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8925
8926  056732  005037  001242                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8927  056736  012737  000012  001122          MOV     #RHDS1,$BDADR   ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
8928  056744  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8929  056750  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8930  056756  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8931  056764  016037  000012  001162          MOV     RHDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8932  056772  013737  001162  001156          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8933  057000  042737  100100  001156          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8934  057006  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8935  057014  016037  000012  001164          MOV     RHDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8936  057022  013737  001164  001160          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8937  057030  042737  100100  001160          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8938  057036  023737  001156  001160          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8939  057044  001006                          BNE     64$             ;BR IF NOT
8940  057046  005737  001156                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8941  057052  001045                          BNE     66$             ;BR IF NOT
8942  057054  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8943  057056  000137  057256                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
8944  057062  013737  001162  001126  64$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8945  057070  013737  001220  001226          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8946  057076  113760  001220  000010          MOVB    PORTB,RHCS2(R0) ;SELECT PORT B.
8947  057104  005737  001156                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8948  057110  001414                          BEQ     65$             ;BR IF ZERO
8949  057112  013737  001216  001226          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8950  057120  013737  001164  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8951  057126  113760  001216  000010          MOVB    PORTA,RHCS2(R0) ;SELECT PORT A.
8952  057134  005737  001160                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8953  057140  001012                          BNE     66$             ;BR IF NOT
```

```
8954  057142  012737  177777  001242  65$:    MOV     #-1,RELERR          ;SET 'RELEASE ERROR' INDICATOR
8955  057150  012760  000011  000000          MOV     #11,RHCS1(R0)       ;CLEAR THE DRIVE
8956  057156  012760  000013  000000          MOV     #13,RHCS1(R0)       ;RELEASE THE DRIVE
8957  057164  104026                          ERROR   26                  ;TYPE ERROR MESSAGE 26
8958  057166  013737  001162  001126  66$:    MOV     $TMP2,$BDDAT        ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
8959  057174  013737  001216  001226          MOV     PORTA,PTNBR         ;CHANGE PORT NUMBER
8960  057202  042737  100000  001162          BIC     #ATA,$TMP2          ;DON'T CHECK THE ATTN BIT
8961  057210  023737  001124  001162          CMP     $GDDAT,$TMP2        ;ALL BITS OK ?
8962  057216  001401                          BEQ     67$                 ;BR IF OK FROM PORT A.
8963  057220  104007                          ERROR   7                   ;REPORT ERROR
8964  057222  013737  001164  001126  67$:    MOV     $TMP3,$BDDAT        ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
8965  057230  013737  001220  001226          MOV     PORTB,PTNBR         ;CHANGE PORT NUMBER
8966  057236  042737  100000  001164          BIC     #ATA,$TMP3          ;DON'T CHECK THE ATTN BIT
8967  057244  023737  001124  001164          CMP     $GDDAT,$TMP3        ;SEE IF READ OK FROM PORT B.
8968  057252  001401                          BEQ     68$                 ;BR IF OK
8969  057254  104007                          ERROR   7                   ;REPORT ERROR
8970  057256  000240                  68$:    NOP
8971  057260  000137  057510                  JMP     $EOP                ;GO TO THE END OF PASS ROUTINE
8972
8973                                  ;;*******************************************************************************
8974
8975                                  .SBTTL  *** SUBROUTINES ***
8976
8977                                  ;;*******************************************************************************
8978
8979
8980                                  ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
8981                                  ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
8982
8983  057264  012737  057334  000004  CKCLK:  MOV     #CKCLK1,@#ERRVEC    ;SET UP VECTOR FOR CLOCK CHECK
8984  057272  005037  000006          CLR     @#ERRVEC+2                  ;NEW PSW
8985  057276  005777  121702          TST     @$LKCSR                     ;CHECK FOR KW11-P
8986  057302  013701  001210          MOV     $LPVEC,R1                   ;KW11-P VECTOR ADDRESS
8987  057306  012721  057416          MOV     #CLOCK,(R1)+                ;SET UP KW11-P VECTOR
8988  057312  012711  000300          MOV     #300,(R1)                   ;PSW - PRI 6
8989  057316  012777  177777  121662  MOV     #-1,@$LKCSB                 ;LOAD COUNTER BUFFER WITH 1'S
8990  057324  012777  000135  121652  MOV     #135,@$LKCSR                ;SET CLOCK - CNT UP, 16MS, CONT INT
8991  057332  000425                  BR      CKCLK3
8992  057334  062706  000004  CKCLK1: ADD     #4,SP                       ;RESTORE THE STACK POINTER
8993  057340  012737  057376  000004  MOV     #CKCLK2,@#ERRVEC            ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
8994  057346  005777  121640          TST     @$LKS                       ;LOOK FOR KW11-L
8995  057352  013701  001214          MOV     $LLVEC,R1                   ;KW11-L VECTOR ADDRESS
8996  057356  012721  057416          MOV     #CLOCK,(R1)+                ;SET UP KW11-L VECTOR
8997  057362  012711  000300          MOV     #300,(R1)                   ;PSW - PRI 6
8998  057366  012777  000100  121616  MOV     #100,@$LKS                  ;SET KW11-L INTERRUPT
8999  057374  000404                  BR      CKCLK3
9000  057376  062706  000004  CKCLK2: ADD     #4,SP                       ;RESTORE THE STACK POINTER
9001  057402  062716  000002          ADD     #2,(SP)                     ;INCREMENT RETURN, NO CLOCK
9002  057406  012737  000006  000004  CKCLK3: MOV     #6,@#ERRVEC         ;RESTORE THE ERROR VECTOR
9003  057414  000207                  RTS     PC
9004
9005                                  ;ROUTINE TO COUNT CLOCK TICKS
9006
9007  057416  062737  000021  001244  CLOCK:  ADD     #17.,TIME           ;ADD 17 MS TO ELAPSED TIME COUNTER
9008  057424  005737  001246          TST     WATCH                       ;IS WATCH ALREADY ZERO ?
9009  057430  001406                  BEQ     1$                          ;BR IF IT IS
```

```
9010  057432  162737  000021  001246          SUB     #17.,WATCH      ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
9011  057440  100002                          BPL     1$              ;BR IF NOT MINUS
9012  057442  005037  001246                  CLR     WATCH           ;CLEAR WATCH DOG COUNTER
9013  057446  000002                  1$:     RTI                     ;RETURN
9014
9015                                  ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
9016
9017  057450  162706  000004          TOLER:  SUB     #4,SP   ;SETUP STACK
9018  057454  016616  000004                  MOV     4(SP),(SP)      ;SAVE STACK
9019  057460  013546                          MOV     @(R5)+,-(SP)    ;GET TIME VALUE
9020  057462  011666  000004                  MOV     (SP),4(SP)      ;MOVE TIME VALUE
9021  057466  011666  000006                  MOV     (SP),6(SP)      ;MOVE VALUE AGAIN
9022  057472  006216                          ASR     (SP)            ;DIVIDE BY 2
9023  057474  006216                          ASR     (SP)            ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
9024  057476  061666  000004                  ADD     (SP),4(SP)      ;CALCULATE UPPER LIMIT FOR TIMEOUT
9025  057502  162666  000004                  SUB     (SP)+,4(SP)     ;CALCULATE LOWER LIMIT FOR TIMEOUT
9026  057506  000205                          RTS     R5              ;RETURN WITH TOLERANCES ON THE STACK
9027
9028                                  ;;**********************************************************************
9029
9030                                  .SBTTL  'SYSMAC' UTILITY ROUTINES
9031
9032                                  ;;**********************************************************************
9033
9034                                  ;*******************************************************************
9035
9036                                  .SBTTL  END OF PASS ROUTINE
9037
9038                                  ;*INCREMENT THE PASS NUMBER ($PASS)
9039                                  ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
9040                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
9041                                  ;*IF THERES A MONITOR GO TO IT
9042                                  ;*IF THERE ISN'T JUMP TO TST1AA
9043
9044  057510                          $EOP:
9045  057510  000004                          SCOPE
9046  057512  005737  001266                  TST     KYBCTL          ;ENTERED TEST VIA KEYBOARD COMMAND ?
9047  057516  001402                          BEQ     .+6             ;BR IF NOT
9048  057520  000137  002410                  JMP     EXEC            ;RETURN TO KEYBOARD CONTROL
9049  057524  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
9050  057530  005037  001170                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
9051  057534  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
9052  057540  042737  100000  001100  .       BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
9053  057546  005327                          DEC     (PC)+           ;;LOOP?
9054  057550  000001                  $EOPCT: .WORD   1
9055  057552  003031                          BGT     $DOAGN          ;;YES
9056  057554  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
9057  057556  000001                  $ENDCT: .WORD   1
9058  057560  057550                          $EOPCT
9059  057562  104400  057642                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
9060  057566  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
9061  057572  104410                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
9062  057574  104400  057657                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
9063  057600  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
9064  057604  001414                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
9065  057606  022700  057626                  CMP     #$ENDAD,R0      ;;IS MONITOR ACT11?
```

```
9066  057612  001005                          BNE     $ENDAD          ;;NO--BRANCH
9067  057614  022760  177777  000002          CMP     #-1,2(R0)       ;;YES--IS THIS THE LAST PASS?
9068  057622  001005                          BNE     $DOAGN          ;;NO--MAKE ANOTHER PASS
9069  057624  000005                          RESET                   ;;CLEAR THE WORLD
9070  057626  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
9071  057630  000240                          NOP                     ;;SAVE ROOM
9072  057632  000240                          NOP                     ;;FOR
9073  057634  000240                          NOP                     ;;ACT11
9074  057636  000137  002704          $DOAGN: JMP     @#TST1AA                ;;RETURN
9075  057642  005015  047105  020104  $ENDMG: .ASCIZ  <15><12>/END PASS #/
9076  057650  040520  051523  021440
9077  057656     000
9078  057657     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
9079                                  ;*****************************************************************
9080
9081                                  .SBTTL  SCOPE HANDLER ROUTINE
9082
9083                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9084                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
9085                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9086                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9087                                  ;*SW14=1         LOOP ON TEST
9088                                  ;*SW11=1         INHIBIT ITERATIONS
9089                                  ;*CALL
9090                                  ;*      SCOPE           ;;SCOPE=IOT
9091
9092  057662                  $SCOPE:
9093  057662  006137  177570          ROL     @#SWR           ;;LOOP ON PRESENT TEST?
9094  057666  100455                  BMI     $OVER           ;;YES IF SW14=1
9095                                  ;#####START OF CODE FOR THE XOR TESTER#####
9096  057670  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
9097                                                          ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
9098  057672  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
9099  057676  012737  057716  000004  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
9100  057704  005737  177060          TST     @#177060        ;;TIME OUT ON XOR?
9101  057710  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
9102  057714  000436                  BR      $SVLAD          ;;GO TO THE NEXT TEST
9103  057716  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
9104  057720  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
9105  057724  000436                  BR      $OVER           ;;LOOP ON THE PRESENT TEST
9106  057726                  6$:;#####END OF CODE FOR THE XOR TESTER#####
9107  057726  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
9108  057732  001404                  BEQ     3$              ;;BR IF NO
9109  057734  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
9110  057740  005037  001170          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
9111  057744  032737  004000  177570  3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
9112  057752  001011                  BNE     1$              ;;BR IF YES
9113  057754  005737  001100          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
9114  057760  001406                  BEQ     1$              ;;      INHIBIT ITERATIONS
9115  057762  005237  001104          INC     $ICNT           ;;INCREMENT ITERATION COUNT
9116  057766  023737  001170  001104  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
9117  057774  002012                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
9118  057776  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
9119  060004  013737  060036  001170          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
9120  060012  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
9121  060016  011637  001106          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

```
9122  060022  013737  001102  177570  $OVER:  MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER
9123  060030  013716  001106                  MOV     $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
9124  060034  000002                          RTI                      ;;FIXES PS
9125  060036  000004                  $MXCNT: 4                        ;;MAX. NUMBER OF ITERATIONS
9126                                  ;*****************************************************************
9127
9128                                  .SBTTL  ERROR HANDLER ROUTINE
9129
9130                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9131                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9132                                  ;*AND GO TO $ERRTYP ON ERROR
9133                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9134                                  ;*SW15=1         HALT ON ERROR
9135                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
9136                                  ;*SW10=1         BELL ON ERROR
9137                                  ;*CALL
9138                                  ;*       ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9139
9140  060040                          $ERROR:
9141  060040  113737  001102  001234          MOVB    $TSTNM,TSTNUM
9142  060046  105237  001103          7$:     INCB    $ERFLG           ;;SET THE ERROR FLAG
9143  060052  001775                          BEQ     7$               ;;DON'T LET THE FLAG GO TO ZERO
9144  060054  013737  001102  177570          MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
9145  060062  032737  002000  177570          BIT     #BIT10,@#SWR     ;;BELL ON ERROR?
9146  060070  001402                          BEQ     1$               ;;NO - SKIP
9147  060072  104400  001174                  TYPE    ,$BELL           ;;RING BELL
9148  060076  005237  001112          1$:     INC     $ERTTL           ;;COUNT THE NUMBER OF ERRORS
9149  060102  011637  001116                  MOV     (SP),$ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
9150  060106  162737  000002  001116          SUB     #2,$ERRPC
9151  060114  117737  120776  001114          MOVB    @$ERRPC,$ITEMB   ;;STRIP AND SAVE THE ERROR ITEM CODE
9152  060122  032737  020000  177570          BIT     #BIT13,@#SWR     ;;SKIP TYPEOUT IF SET
9153  060130  001004                          BNE     2$               ;;SKIP TYPEOUTS
9154  060132  004737  060154                  JSR     PC,@#$ERRTYP     ;;GO TO USER ERROR ROUTINE
9155  060136  104400  001201                  TYPE    ,$CRLF
9156  060142  005737  177570          2$:     TST     @#SWR            ;;HALT ON ERROR
9157  060146  100001                          BPL     3$               ;;SKIP IF CONTINUE
9158  060150  000000                          HALT                     ;;HALT ON ERROR!
9159  060152                          3$:
9160  060152  000002                          RTI                      ;;RETURN
9161                                  ;*****************************************************************
9162
9163                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
9164
9165                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9166                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
9167                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
9168
9169  060154                          $ERRTYP:
9170  060154  104400  001201                  TYPE    ,$CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
9171  060160  010046                          MOV     R0,-(SP)         ;;SAVE R0
9172  060162  005000                          CLR     R0               ;;PICKUP THE ITEM INDEX
9173  060164  153700  001114                  BISB    @#$ITEMB,R0
9174  060170  001004                          BNE     1$               ;;IF ITEM NUMBER IS ZERO, JUST
9175                                                                   ;;TYPE THE PC OF THE ERROR
9176  060172  013746  001116                  MOV     $ERRPC,-(SP)     ;;SAVE $ERRPC FOR TYPEOUT
9177                                                                   ;;ERROR ADDRESS
```

```
 9178  060176  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9179  060200  000445                    BR      10$             ;;GET OUT
 9180  060202  005300              1$:   DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 9181  060204  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
 9182  060206  006300                    ASL     R0
 9183  060210  006300                    ASL     R0
 9184  060212  062700  001274            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 9185  060216  012037  060226            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 9186  060222  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 9187  060224  104400                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
 9188  060226  000000              2$:   .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 9189  060230  104400  001201            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9190  060234  012037  060244      3$:   MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 9191  060240  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
 9192  060242  104400                    TYPE                    ;;TYPE THE "DATA HEADER"
 9193  060244  000000              4$:   .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 9194  060246  104400  001201            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9195  060252  010146              5$:   MOV     R1,-(SP)        ;;SAVE R1
 9196  060254  012001                    MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
 9197  060256  001415                    BEQ     9$              ;;BR IF NO DATA TO BE TYPED
 9198  060260  012000                    MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
 9199  060262  105720              6$:   TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
 9200  060264  001003                    BNE     7$              ;;BR IF DECIMAL
 9201  060266  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 9202  060270  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9203  060272  000402                    BR      8$
 9204  060274                      7$:
 9205  060274  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 9206  060276  104410                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 9207  060300  005711              8$:   TST     (R1)            ;;IS THERE ANOTHER NUMBER?
 9208  060302  001403                    BEQ     9$              ;;BR IF NO
 9209  060304  104400  060324            TYPE    ,11$            ;;TYPE TWO(2) SPACES
 9210  060310  000764                    BR      6$              ;;LOOP
 9211
 9212  060312  012601              9$:   MOV     (SP)+,R1        ;;RESTORE R1
 9213  060314  012600              10$:  MOV     (SP)+,R0        ;;RESTORE R0
 9214  060316  104400  001201            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 9215  060322  000207                    RTS     PC              ;;RETURN
 9216  060324  020040      000     11$:  .ASCIZ  /  /            ;;TWO(2) SPACES
 9217          060330                    .EVEN
 9218                              ;*****************************************************************
 9219
 9220                              .SBTTL  TYPE ROUTINE
 9221
 9222                              ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 9223                              ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 9224                              ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 9225                              ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 9226                              ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 9227                              ;*
 9228                              ;*CALL:
 9229                              ;*1) USING A TRAP INSTRUCTION
 9230                              ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 9231                              ;*OR
 9232                              ;*      TYPE
 9233                              ;*      MESADR
```

```
                                        ;*
                                        ;*2) USING A JSR INSTRUCTION
                                        ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
                                        ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                        ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
                                        
060330  105737  001151          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
060334  100002                          BPL     1$              ;;BR IF YES
060336  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
060340  000407                          BR      3$              ;;LEAVE
060342  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
060344  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
060350  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
060352  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
060354  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
060356  012600                          MOV     (SP)+,R0        ;;RESTORE R0
060360  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
060364  000002                          RTI                     ;;RETURN
060366  004737  060420          4$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
060372  123726  001150          5$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
060376  001364                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
060400  013746  001146                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
060404  105366  000001          6$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
060410  002770                          BLT     5$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
060412  004737  060420                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
060416  000772                          BR      6$              ;;LOOP
060420  105777  120516          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
060424  100375                          BPL     $TYPEC
060426  116677  000002  120510          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
060434  000207                          RTS     PC
                                        ;******************************************************************
                                        
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
                                        
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M=1 OR 0
                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
```

```
9290
9291  060436 017646 000000        $TYPOS: MOV   @(SP),-(SP)      ;;PICKUP THE MODE
9292  060442 116637 000001 060661         MOVB  1(SP),$OFILL     ;;LOAD ZERO FILL SWITCH
9293  060450 112637 060663                MOVB  (SP)+,$OMODE+1   ;;NUMBER OF DIGITS TO TYPE
9294  060454 062716 000002                ADD   #2,(SP)          ;;ADJUST RETURN ADDRESS
9295  060460 000406                       BR    $TYPON
9296  060462 112737 000001 060661 $TYPOC: MOVB  #1,$OFILL        ;;SET THE ZERO FILL SWITCH
9297  060470 112737 000006 060663         MOVB  #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
9298  060476 112737 000005 060660 $TYPON: MOVB  #5,$OCNT         ;;SET THE ITERATION COUNT
9299  060504 010346                       MOV   R3,-(SP)         ;;SAVE R3
9300  060506 010446                       MOV   R4,-(SP)         ;;SAVE R4
9301  060510 010546                       MOV   R5,-(SP)         ;;SAVE R5
9302  060512 113704 060663                MOVB  $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
9303  060516 005404                       NEG   R4
9304  060520 062704 000006                ADD   #6,R4            ;;SUBTRACT IT FOR MAX. ALLOWED
9305  060524 110437 060662                MOVB  R4,$OMODE        ;;SAVE IT FOR USE
9306  060530 113704 060661                MOVB  $OFILL,R4        ;;GET THE ZERO FILL SWITCH
9307  060534 016605 000012                MOV   12(SP),R5        ;;PICKUP THE INPUT NUMBER
9308  060540 005003                       CLR   R3               ;;CLEAR THE OUTPUT WORD
9309  060542 006105               1$:     ROL   R5               ;;ROTATE MSB INTO "C"
9310  060544 000404                       BR    3$               ;;GO DO MSB
9311  060546 006105               2$:     ROL   R5               ;;FORM THIS DIGIT
9312  060550 006105                       ROL   R5
9313  060552 006105                       ROL   R5
9314  060554 010503                       MOV   R5,R3
9315  060556 006103               3$:     ROL   R3               ;;GET LSB OF THIS DIGIT
9316  060560 105337 060662                DECB  $OMODE           ;;TYPE THIS DIGIT?
9317  060564 100016                       BPL   7$               ;;BR IF NO
9318  060566 042703 177770                BIC   #177770,R3       ;;GET RID OF JUNK
9319  060572 001002                       BNE   4$               ;;TEST FOR 0
9320  060574 005704                       TST   R4               ;;SUPPRESS THIS 0?
9321  060576 001403                       BEQ   5$               ;;BR IF YES
9322  060600 005204               4$:     INC   R4               ;;DON'T SUPPRESS ANYMORE 0'S
9323  060602 052703 000060                BIS   #'0,R3           ;;MAKE THIS DIGIT ASCII
9324  060606 052703 000040        5$:     BIS   #' ,R3           ;;MAKE ASCII IF NOT ALREADY
9325  060612 110337 060656                MOVB  R3,8$            ;;SAVE FOR TYPING
9326  060616 104400 060656                TYPE  ,8$              ;;GO TYPE THIS DIGIT
9327  060622 105337 060660        7$:     DECB  $OCNT            ;;COUNT BY 1
9328  060626 003347                       BGT   2$               ;;BR IF MORE TO DO
9329  060630 002402                       BLT   6$               ;;BR IF DONE
9330  060632 005204                       INC   R4               ;;INSURE LAST DIGIT ISN'T A BLANK
9331  060634 000744                       BR    2$               ;;GO DO THE LAST DIGIT
9332  060636 012605               6$:     MOV   (SP)+,R5         ;;RESTORE R5
9333  060640 012604                       MOV   (SP)+,R4         ;;RESTORE R4
9334  060642 012603                       MOV   (SP)+,R3         ;;RESTORE R3
9335  060644 016666 000002 000004         MOV   2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
9336  060652 012616                       MOV   (SP)+,(SP)
9337  060654 000002                       RTI                    ;;RETURN
9338  060656    000               8$:     .BYTE 0                ;;STORAGE FOR ASCII DIGIT
9339  060657    000                       .BYTE 0                ;;TERMINATOR FOR TYPE ROUTINE
9340  060660    000               $OCNT:  .BYTE 0                ;;OCTAL DIGIT COUNTER
9341  060661    000               $OFILL: .BYTE 0                ;;ZERO FILL SWITCH
9342  060662 000000               $OMODE: .WORD 0                ;;NUMBER OF DIGITS TO TYPE
9343                              ;******************************************************************
9344
9345                              .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
```

```
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                                        $TYPDS:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
                                                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
                                                BPL     1$              ;;BR IF INPUT IS POS.
                                                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
                                                MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
                                        1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
                                                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
                                                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
                                        2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
                                                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
                                        3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
                                                BLT     4$              ;;BR IF DONE
                                                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
                                                BR      3$
                                        4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
                                                TST     R2              ;;CHECK IF BCD DIGIT=0
                                                BNE     5$              ;;FALL THROUGH IF 0
                                                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
                                                BMI     7$              ;;BR IF YES
                                        5$:     ASLB    (SP)            ;;MSD?
                                                BCC     6$              ;;BR IF NO
                                                MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
                                        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
                                        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
                                                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                                                TST     (R0)+           ;;JUST INCREMENTING
                                                CMP     R0,#10          ;;CHECK THE TABLE INDEX
                                                BLT     2$              ;;GO DO THE NEXT DIGIT
                                                BGT     8$              ;;GO TO EXIT
                                                MOV     R5,R2           ;;GET THE LSD
                                                BR      6$              ;;GO CHANGE TO ASCII
                                        8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                                                BPL     9$              ;;BR IF NO
                                                MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                                        9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                                                MOV     (SP)+,R5        ;;POP STACK INTO R5
                                                MOV     (SP)+,R3        ;;POP STACK INTO R3
                                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
 9402  061052  104400  061100                TYPE    $DBLK           ;;NOW TYPE THE NUMBER
 9403  061056  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 9404  061064  012616                        MOV     (SP)+,(SP)
 9405  061066  000002                        RTI                     ;;RETURN TO USER
 9406  061070  023420                $DTBL:  10000.
 9407  061072  001750                        1000.
 9408  061074  000144                        100.
 9409  061076  000012                        10.
 9410  061100  000004                $DBLK:  .BLKW   4
 9411                                 ;*****************************************************************
 9412
 9413                                 .SBTTL  TTY INPUT ROUTINE
 9414
 9415                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 9416                                 ;*CALL:
 9417                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 9418                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 9419                                 ;
 9420
 9421  061110  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 9422  061112  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
 9423  061120  105777  120012        1$:     TSTB    @$TKS           ;;WAIT FOR
 9424  061124  100375                        BPL     1$              ;;A CHARACTER
 9425  061126  117766  120006  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
 9426  061134  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 9427  061142  000002                        RTI                     ;;GO BACK TO USER
 9428                                 ;*****************************************************************
 9429                                 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 9430                                 ;*CALL:
 9431                                 ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 9432                                 ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 9433                                 ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 9434
 9435  061144  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 9436  061146  005046                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
 9437  061150  012703  061405        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 9438  061154  022703  061414        2$:     CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
 9439  061160  101456                        BLOS    4$              ;;BR IF YES
 9440  061162  104412                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 9441  061164  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
 9442  061166  122713  000177              CMPB    #177,(R3)       ;;IS IT A RUBOUT
 9443  061172  001022                        BNE     5$              ;;BR IF NO
 9444  061174  005716                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
 9445  061176  001007                        BNE     6$              ;;BR IF NO
 9446  061200  112737  000134  061376        MOVB    #'\,9$          ;;TYPE A BACK SLASH
 9447  061206  104400  061376                TYPE    ,9$
 9448  061212  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
 9449  061216  005303                6$:     DEC     R3              ;;BACKUP BY ONE
 9450  061220  020327  061405                CMP     R3,#$TTYIN      ;;STACK EMPTY?
 9451  061224  103434                        BLO     4$              ;;BR IF YES
 9452  061226  111337  061376                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
 9453  061232  104400  061376                TYPE    ,9$             ;;GO TYPE
 9454  061236  000746                        BR      2$              ;;GO READ ANOTHER CHAR.
 9455  061240  005716                5$:     TST     (SP)            ;;RUBOUT KEY SET?
 9456  061242  001406                        BEQ     7$              ;;BR IF NO
 9457  061244  112737  000134  061376        MOVB    #'\,9$          ;;TYPE A BACK SLASH
```

```
9458  061252  104400  061376                  TYPE    9$
9459  061256  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
9460  061260  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
9461  061264  001003                          BNE     8$              ;;BR IF NO
9462  061266  104400  061400                  TYPE    $CNTLU          ;;TYPE A CONTROL "U"
9463  061272  000726                          BR      1$              ;;GO START OVER
9464  061274  122713  000012          8$:     CMPB    #12,(R3)        ;;IS CHARACTER A "LF"?
9465  061300  001011                          BNE     3$              ;;BRANCH IF NO
9466  061302  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
9467  061304  104400  001201                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
9468  061310  104400  061405                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
9469  061314  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
9470  061316  104400  001200          4$:     TYPE    ,$QUES          ;;TYPE A '?'
9471  061322  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
9472  061324  111337  061376          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
9473  061330  104400  061376                  TYPE    ,9$
9474  061334  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
9475  061340  001305                          BNE     2$              ;;LOOP IF NOT RETURN
9476  061342  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
9477  061346  104400  001202                  TYPE    ,$LF            ;;TYPE A LINE FEED
9478  061352  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
9479  061354  012603                          MOV     (SP)+,R3        ;;RESTORE R3
9480  061356  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
9481  061360  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
9482  061366  012766  061405  000004          MOV     #$TTYIN,4(SP)
9483  061374  000002                          RTI                     ;;RETURN
9484  061376     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
9485  061377     000                  .BYTE   0                       ;;TERMINATOR
9486  061400  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
9487  061405  000007                  $TTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
9488                                  ;*****************************************************************
9499
9490                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
9491
9492                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9493                                  ;*CHANGE IT TO BINARY.
9494                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9495                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
9496                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9497                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9498                                  ;*CALL:
9499                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
9500                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
9501                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
9502
9503  061414  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
9504  061416  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
9505  061424  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9506  061426  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9507  061430  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9508  061432  104414                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
9509  061434  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
9510  061436  010037  061542                  MOV     R0,5$           ;;AND SAVE IT
9511  061442  005001                          CLR     R1              ;;CLEAR DATA WORD
9512  061444  005002                          CLR     R2
9513  061446  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
```

```
9514  061450  001420                        BEQ     3$                ;;IF ZERO GET OUT
9515  061452  122716  000060                CMPB    #'0,(SP)          ;;MAKE SURE THIS CHARACTER
9516  061456  003026                        BGT     4$                ;;IS AN OCTAL DIGIT
9517  061460  122716  000067                CMPB    #'7,(SP)
9518  061464  002423                        BLT     4$
9519  061466  006301                        ASL     R1                ;;*2
9520  061470  006102                        ROL     R2
9521  061472  006301                        ASL     R1                ;;*4
9522  061474  006102                        ROL     R2
9523  061476  006301                        ASL     R1                ;;*8
9524  061500  006102                        ROL     R2
9525  061502  042716  177770                BIC     #^C7,(SP)         ;;STRIP THE ASCII JUNK
9526  061506  062601                        ADD     (SP)+,R1          ;;ADD IN THIS DIGIT
9527  061510  000756                        BR      2$                ;;LOOP
9528  061512  005726                3$:     TST     (SP)+             ;;CLEAN TERMINATOR FROM STACK
9529  061514  010166  000012                MOV     R1,12(SP)         ;;SAVE THE RESULT
9530  061520  010237  061552                MOV     R2,$HIOCT
9531  061524  012602                        MOV     (SP)+,R2          ;;POP STACK INTO R2
9532  061526  012601                        MOV     (SP)+,R1          ;;POP STACK INTO R1
9533  061530  012600                        MOV     (SP)+,R0          ;;POP STACK INTO R0
9534  061532  000002                        RTI                       ;;RETURN
9535  061534  005726                4$:     TST     (SP)+             ;;CLEAN PARTIAL FROM STACK
9536  061536  105010                        CLRB    (R0)              ;;SET A TERMINATOR
9537  061540  104400                        TYPE                      ;;TYPE UP THRU THE BAD CHAR.
9538  061542  000000                5$:     .WORD   0
9539  061544  104400  001200                TYPE    ,$QUES            ;;"?" "CR" & "LF"
9540  061550  000730                        BR      1$                ;;TRY AGAIN
9541  061552  000000                $HIOCT: .WORD   0                 ;;HIGH ORDER BITS GO HERE
9542                                ;******************************************************************
9543
9544                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
9545
9546                                ;*SAVE R0-R5
9547                                ;*CALL:
9548                                ;*      SAVREG
9549                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
9550                                ;*
9551                                ;*TOP---(+16)
9552                                ;* +2---(+18)
9553                                ;* +4---R5
9554                                ;* +6---R4
9555                                ;* +8---R3
9556                                ;*+10---R2
9557                                ;*+12---R1
9558                                ;*+14---R0
9559
9560  061554                        $SAVREG:
9561  061554  010046                        MOV     R0,-(SP)          ;;PUSH R0 ON STACK
9562  061556  010146                        MOV     R1,-(SP)          ;;PUSH R1 ON STACK
9563  061560  010246                        MOV     R2,-(SP)          ;;PUSH R2 ON STACK
9564  061562  010346                        MOV     R3,-(SP)          ;;PUSH R3 ON STACK
9565  061564  010446                        MOV     R4,-(SP)          ;;PUSH R4 ON STACK
9566  061566  010546                        MOV     R5,-(SP)          ;;PUSH R5 ON STACK
9567  061570  016646  000022                MOV     22(SP),-(SP)      ;;SAVE PS OF MAIN FLOW
9568  061574  016646  000022                MOV     22(SP),-(SP)      ;;SAVE PC OF MAIN FLOW
9569  061600  016646  000022                MOV     22(SP),-(SP)      ;;SAVE PS OF CALL
```

```
9570  061604  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
9571  061610  000002                          RTI
9572
9573                                  ;*RESTORE R0-R5
9574                                  ;*CALL:
9575                                  ;*      RESREG
9576  061612                          $RESREG:
9577  061612  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
9578  061616  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
9579  061622  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
9580  061626  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
9581  061632  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
9582  061634  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
9583  061636  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
9584  061640  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
9585  061642  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9586  061644  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9587  061646  000002                          RTI
9588                                  ;*****************************************************************
9589
9590                                  .SBTTL  TRAP DECODER
9591
9592                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
9593                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
9594                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
9595                                  ;*GO TO THAT ROUTINE.
9596
9597  061650  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
9598  061652  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
9599  061656  005740                          TST     -(R0)           ;;BACKUP BY 2
9600  061660  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
9601  061662  016000  061670                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
9602  061666  000200                          RTS     R0              ;;GO TO ROUTINE
9603
9604
9605                                  .SBTTL  TRAP TABLE
9606
9607                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
9608                                  ;*BY THE "TRAP" INSTRUCTION.
9609
9610                                  ;       ROUTINE
9611                                  ;       -------
9612  061670                          $TRPAD:
9613  061670  060330                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
9614  061672  060462                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
9615  061674  060436                          $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
9616  061676  060476                          $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
9617  061700  060664                          $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
9618  061702  061110                          $RDCHR  ;;CALL=RDCHR    TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE
9619  061704  061144                          $RDLIN  ;;CALL=RDLIN    TRAP+14(104414) TTY TYPEIN STRING ROUTINE
9620  061706  061414                          $RDOCT  ;;CALL=RDOCT    TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY
9621  061710  061554                          $SAVREG ;;CALL=SAVREG   TRAP+20(104420) SAVE R0-R5 ROUTINE
9622  061712  061612                          $RESREG ;;CALL=RESREG   TRAP+22(104422) RESTORE R0-R5 ROUTINE
9623
9624                                  ;;*****************************************************************
9625
```

```
                                        .SBTTL  TELETYPE MESSAGES
                                        ;;**************************************************************************
061714  005015  042115  030455  TITLE:  .ASCII  <15><12>/MD-11-DERPP-A/<15><12><12>
061722  026461  042504  050122
061730  026520  006501  005012
061736  050122  032060  042040        .ASCIZ  /RP04 DUAL CONTROLLER LOGIC TEST - PART 1/<15><12><12>
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
062012  005015  047105  042524  ENTERA: .ASCIZ  <15><12>/ENTER DRIVE ADDRESS: /
062020  020122  051104  053111
062026  020105  042101  051104
062034  051505  035123  000040
062042  047111  040526  044514  ADRERR: .ASCIZ  /INVALID ADDRESS/<15><12>
062050  020104  042101  051104
062056  051505  006523  000012
062064  005015  047520  052122  PORTAIS: .ASCIZ <15><12>/PORT A ADDRESS IS: /
062072  040440  040440  042104
062100  042522  051523  044440
062106  035123  000040
062112  005015  047520  052122  PORTBIS: .ASCIZ <15><12>/PORT B ADDRESS IS: /
062120  041040  040440  042104
062126  042522  051523  044440
062134  035123  000040
062140  005015  054523  052123  NOCLOCK: .ASCIZ <15><12>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<15><12><12>
062146  046505  046440  051525
062154  020124  040510  042526
062162  023440  023514  047440
062170  020122  050047  020047
062176  046103  041517  006513
062204  005012     000
062207     012  047105  042524  TESTNO: .ASCIZ  <12>/ENTER TEST #: /
062214  020122  042524  052123
062222  021440  020072     000
062227     111  053116  046101  BADNO:  .ASCIZ  /INVALID TEST NUMBER/<15><12>
062234  042111  052040  051505
062242  020124  052516  041115
062250  051105  005015     000
062255     040  051105  047522  TSTERR: .ASCIZ  / ERRORS/<15><12>
062262  051522  005015     000
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
062347     012  047105  042524  NTRH11: .ASCIZ  <12>/ENTER NEW RH11 ADDRESS: /
```

```
 9682  062354  020122  042516  020127
 9683  062362  044122  030461  040440
 9684  062370  042104  042522  051523
 9685  062376  020072     000
 9686  062401     015  052012  042510  NORESP: .ASCIZ  <15><12>/THE RH11 DID NOT RESPOND WHEN RHWC ACCESSED AT ADDR: /
 9687  062406  051040  030510  020061
 9688  062414  044504  020104  047516
 9689  062422  020124  042522  050123
 9690  062430  047117  020104  044127
 9691  062436  047105  051040  053510
 9692  062444  020103  041501  042503
 9693  062452  051523  042105  040440
 9694  062460  020124  042101  051104
 9695  062466  020072     000
 9696
 9697                          ;;*****************************************************************************
 9698
 9699                          .SBTTL  TEST ERROR MESSAGES
 9700
 9701                          ;;*****************************************************************************
 9702
 9703  062471     127  047522  043516  EM1:    .ASCIZ  /WRONG DRIVE TYPE/
 9704  062476  042040  044522  042526
 9705  062504  052040  050131  000105
 9706
 9707  062512  051104  053111  020105  EM2:    .ASCIZ  /DRIVE NOT ON LINE/
 9708  062520  047516  020124  047117
 9709  062526  046040  047111  000105
 9710
 9711  062534  042523  044522  046101  EM3:    .ASCIZ  /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
 9712  062542  047040  046525  042502
 9713  062550  020122  042522  042101
 9714  062556  052040  051110  052517
 9715  062564  044107  042440  041501
 9716  062572  020110  047520  052122
 9717  062600  047040  052117  052040
 9718  062606  042510  051440  046501
 9719  062614  000105
 9720
 9721  062616  051104  053111  020105  EM4:    .ASCIZ  /DRIVE NOT SEIZED BY PORT/
 9722  062624  047516  020124  042523
 9723  062632  055111  042105  041040
 9724  062640  020131  047520  052122
 9725  062646     000
 9726
 9727  062647     127  047522  043516  EM5:    .ASCIZ  /WRONG STATUS SEEN BY THE SEIZING PORT/
 9728  062654  051440  040524  052524
 9729  062662  020123  042523  047105
 9730  062670  041040  020131  044124
 9731  062676  020105  042523  055111
 9732  062704  047111  020107  047520
 9733  062712  052122     000
 9734
 9735  062715     122  043505  051511  EM6:    .ASCIZ  /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
 9736  062722  042524  020122  047503
 9737  062730  052116  047105  051524
```

```
 9738  062736  053440  051105  020105
 9739  062744  042523  047105  041040
 9740  062752  020131  050117  047520
 9741  062760  044523  042524  050040
 9742  062766  051117  020124  020055
 9743  062774  051104  053111  020105
 9744  063002  040527  020123  042523
 9745  063010  055111  042105     000
 9746
 9747  063015     122  043505  051511  EM7:    .ASCIZ  /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
 9748  063022  042524  020122  047503
 9749  063030  052116  047105  051524
 9750  063036  053440  047522  043516
 9751  063044  040440  052106  051105
 9752  063052  051040  046105  040505
 9753  063060  042523  047440  020122
 9754  063066  044524  042515  052517
 9755  063074  000124
 9756
 9757  063076  042522  044507  052123  EM10:   .ASCIZ  /REGISTER CONTENTS WRONG/
 9758  063104  051105  041440  047117
 9759  063112  042524  052116  020123
 9760  063120  051127  047117  000107
 9761
 9762  063126  047503  052116  047522  EM11:   .ASCIZ  /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
 9763  063134  020114  052502  020123
 9764  063142  040520  044522  054524
 9765  063150  042440  051122  051117
 9766  063156  051040  040505  044504
 9767  063164  043516  044440  042116
 9768  063172  041511  052101  042105
 9769  063200  051040  043505  051511
 9770  063206  042524  000122
 9771
 9772  063212  051104  053111  020105  EM12:   .ASCIZ  /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
 9773  063220  047516  020124  042523
 9774  063226  055111  042105  041040
 9775  063234  020131  051104  053111
 9776  063242  020105  046103  040505
 9777  063250  020122  047503  046515
 9778  063256  047101  000104
 9779
 9780  063262  042522  042101  047111  EM13:   .ASCIZ  /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
 9781  063270  050040  042522  042523
 9782  063276  020124  047504  051505
 9783  063304  047040  052117  051440
 9784  063312  052105  053040  046117
 9785  063320  046525  020105  040526
 9786  063326  044514  020104  047506
 9787  063334  020122  044124  020105
 9788  063342  047520  052122     000
 9789
 9790  063347     126  046117  046525  EM14:   .ASCIZ  /VOLUME VALID SET ON THE WRONG PORT/
 9791  063354  020105  040526  044514
 9792  063362  020104  042523  020124
 9793  063370  047117  052040  042510
```

```
 9794  063376  053440  047522  043516
 9795  063404  050040  051117  000124
 9796
 9797  063412  052101  047124  041040  EM15:   .ASCIZ  /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
 9798  063420  052111  053440  047522
 9799  063426  043516  040440  052106
 9800  063434  051105  052040  046511
 9801  063442  047505  052125  026440
 9802  063450  051040  050505  042525
 9803  063456  052123  047040  052117
 9804  063464  051440  052105     000
 9805
 9806  063471     101  052124  020116  EM16:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
 9807  063476  044502  020124  051127
 9808  063504  047117  020107  043101
 9809  063512  042524  020122  042522
 9810  063520  042514  051501  020105
 9811  063526  020055  042522  052521
 9812  063534  051505  020124  042523
 9813  063542  000124
 9814
 9815  063544  052101  047124  041040  EM17:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
 9816  063552  052111  053440  047522
 9817  063560  043516  040440  052106
 9818  063566  051105  051040  046105
 9819  063574  040505  042523  026440
 9820  063602  051040  050505  042525
 9821  063610  052123  047040  052117
 9822  063616  051440  052105     000
 9823
 9824  063623     104  044522  042526  EM20:   .ASCIZ  /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
 9825  063630  047040  052117  051440
 9826  063636  044505  042532  020104
 9827  063644  044127  047105  040440
 9828  063652  052124  020116  044502
 9829  063660  020124  047506  020122
 9830  063666  047520  052122  041440
 9831  063674  042514  051101  042105
 9832  063702     000
 9833
 9834  063703     104  044522  042526  EM21:   .ASCIZ  /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
 9835  063710  051440  044505  042532
 9836  063716  020104  044127  047105
 9837  063724  055040  051105  020117
 9838  063732  051127  052111  042524
 9839  063740  020116  047111  040440
 9840  063746  052124  020116  044502
 9841  063754  000124
 9842
 9843  063756  051104  053111  020105  EM22:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
 9844  063764  047516  020124  047111
 9845  063772  047040  052505  051124
 9846  064000  046101  040440  052106
 9847  064006  051105  052040  046511
 9848  064014  047505  052125  026440
 9849  064022  051040  050505  042525
```

```
 9850  064030  052123  047040  052117
 9851  064036  051440  052105     000
 9852
 9853  064043     124  046511  047505  EM23:   .ASCIZ  /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
 9854  064050  052125  041440  042514
 9855  064056  051101  042105  052040
 9856  064064  042510  042040  044522
 9857  064072  042526  051447  042440
 9858  064100  051122  051117  041040
 9859  064106  052111     000
 9860
 9861  064111     122  046105  040505  EM24:   .ASCIZ  /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
 9862  064116  042523  041440  046517
 9863  064124  040515  042116  051040
 9864  064132  046105  040505  042523
 9865  064140  020104  051104  053111
 9866  064146  020105  044527  044124
 9867  064154  042440  051122  051117
 9868  064162  020123  042523  000124
 9869
 9870  064170  044524  042515  052517  EM25:   .ASCIZ  /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
 9871  064176  020124  047117  026505
 9872  064204  044123  052117  042040
 9873  064212  042111  047040  052117
 9874  064220  051040  052105  044522
 9875  064226  043507  051105     000
 9876
 9877  064233     104  044522  042526  EM26:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
 9878  064240  047040  052117  044440
 9879  064246  020116  042516  052125
 9880  064254  040522  020114  043101
 9881  064262  042524  020122  042522
 9882  064270  042514  051501  020105
 9883  064276  020055  042522  052521
 9884  064304  051505  020124  047516
 9885  064312  020124  042523  000124
 9886
 9887  064320  042522  044507  052123  EM27:   .ASCIZ  /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
 9888  064326  051105  053440  047522
 9889  064334  043516  040440  052106
 9890  064342  051105  051040  046105
 9891  064350  040505  042523  053440
 9892  064356  052111  020110  042522
 9893  064364  052521  051505  020124
 9894  064372  042523  000124
 9895
 9896  064376  051104  053111  020105  EM30:   .ASCIZ  /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
 9897  064404  042523  055111  042105
 9898  064412  041040  020131  042522
 9899  064420  042514  051501  020105
 9900  064426  047503  046515  047101
 9901  064434  020104  051511  052523
 9902  064442  042105  053440  042510
 9903  064450  020116  051104  053111
 9904  064456  020105  047111  047040
 9905  064464  052505  051124  046101
```

```
9906  064472     000
9907
9908  064473     104  044522  042526  EM31:   .ASCIZ  /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
9909  064500  044440  020116  042516
9910  064506  052125  040522  020114
9911  064514  043101  042524  020122
9912  064522  042522  042514  051501
9913  064530  020105  020055  042522
9914  064536  052521  051505  020124
9915  064544  042523  000124
9916
9917  064550  052101  047124  041040  EM32:   .ASCIZ  /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
9918  064556  052111  053440  047522
9919  064564  043516  040440  052106
9920  064572  051105  051040  041505
9921  064600  046101  041111  040522
9922  064606  042524  041440  046517
9923  064614  040515  042116     000
9924
9925  064621     104  044522  042526  EM33:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
9926  064626  051040  052105  051125
9927  064634  042516  020104  047524
9928  064642  047040  052505  051124
9929  064650  046101  044440  020106
9930  064656  051104  053111  020105
9931  064664  046103  040505  020122
9932  064672  044507  042526  020116
9933  064700  044127  046111  020105
9934  064706  051104  053111  020105
9935  064714  042523  055111  042105
9936  064722     000
9937
9938  064723     104  044522  042526  EM34:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
9939  064730  051040  052105  051125
9940  064736  042516  020104  047524
9941  064744  047040  052505  051124
9942  064752  046101  044440  020106
9943  064760  040515  051523  052502
9944  064766  020123  047111  052111
9945  064774  043440  053111  047105
9946  065002  053440  044510  042514
9947  065010  042040  044522  042526
9948  065016  051440  044505  042532
9949  065024  000104
9950
9951  065026  044524  042515  052517  EM35:   .ASCIZ  /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
9952  065034  020124  047117  020105
9953  065042  044123  052117  043040
9954  065050  051111  042105  053440
9955  065056  052111  047510  052125
9956  065064  051040  043505  051511
9957  065072  042524  020122  041501
9958  065100  042503  051523     000
9959
9960  065105     124  046511  047505  EM36:   .ASCIZ  /TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS/
9961  065112  052125  044040  051501
```

```
 9962  065120  047040  052117  047440
 9963  065126  041503  051125  042105
 9964  065134  053440  052111  044510
 9965  065142  020116  020062  042523
 9966  065150  047503  042116  000123
 9967
 9968  065156  051104  053111  020105  EM37:   .ASCIZ  /DRIVE IS NON-EXISTANT ('NED' BIT SET)/
 9969  065164  051511  047040  047117
 9970  065172  042455  044530  052123
 9971  065200  047101  020124  023450
 9972  065206  042516  023504  041040
 9973  065214  052111  051440  052105
 9974  065222  000051
 9975
 9976  065224  052101  047124  041040  EM40:   .ASCIZ  /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
 9977  065232  052111  043040  051117
 9978  065240  050040  051117  020124
 9979  065246  047516  020124  042522
 9980  065254  042523  020124  054502
 9981  065262  046440  051501  041123
 9982  065270  051525  041440  042514
 9983  065276  051101     000
 9984
 9985  065301     124  046511  047505  EM41:   .ASCIZ  /TIMEOUT CLEARED THE ATTENTION BIT/
 9986  065306  052125  041440  042514
 9987  065314  051101  042105  052040
 9988  065322  042510  040440  052124
 9989  065330  047105  044524  047117
 9990  065336  041040  052111     000
 9991
 9992  065343     104  044522  042526  EM42:   .ASCIZ  /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
 9993  065350  047040  052117  044440
 9994  065356  020116  042516  052125
 9995  065364  040522  020114  051117
 9996  065372  051440  044505  042532
 9997  065400  020104  043101  042524
 9998  065406  020122  052101  047124
 9999  065414  041040  052111  053440
10000  065422  044522  052124  047105
10001  065430     000
10002
10003  065431     104  044522  042526  EM43:   .ASCIZ  /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
10004  065436  044440  020116  042516
10005  065444  052125  040522  020114
10006  065452  043101  042524  020122
10007  065460  052101  042524  052116
10008  065466  047511  020116  044502
10009  065474  020124  051127  052111
10010  065502  042524  000116
10011
10012  065506  051127  052111  020105  EM44:   .ASCIZ  /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
10013  065514  052101  042524  052116
10014  065522  047511  020116  044502
10015  065530  020124  044504  020104
10016  065536  047516  020124  042523
10017  065544  020124  047520  052122
```

```
10018  065552  051040  050505  042525
10019  065560  052123     000
10020
10021  065563     103  047117  051124  EM45:   .ASCIZ  @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
10022  065570  046117  042514  020122
10023  065576  042523  042514  052103
10024  065604  051440  044527  041524
10025  065612  020110  047117  042040
10026  065620  044522  042526  047040
10027  065626  052117  044440  020116
10028  065634  040447  041057  000047
10029
10030  065642  040503  023516  020124  EM46:   .ASCIZ  /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
10031  065650  041501  042503  051523
10032  065656  042040  044522  042526
10033  065664  052040  051110  052517
10034  065672  044107  042440  052111
10035  065700  042510  020122  047520
10036  065706  052122     000
10037
10038  065711     101  052124  020116  EM47:   .ASCIZ  /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
10039  065716  044502  020124  047506
10040  065724  020122  042523  055111
10041  065732  047111  020107  047520
10042  065740  052122  047040  052117
10043  065746  041440  042514  051101
10044  065754  042105  041040  020131
10045  065762  040515  051523  052502
10046  065770  020123  047111  052111
10047  065776     000
10048
10049  065777     101  052124  020116  EM50:   .ASCIZ  /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
10050  066004  044502  020124  047506
10051  066012  020122  050117  047520
10052  066020  044523  042524  050040
10053  066026  051117  020124  046103
10054  066034  040505  042522  020104
10055  066042  054502  046440  051501
10056  066050  041123  051525  044440
10057  066056  044516  000124
10058
10059  066062  052101  047124  041040  EM51:   .ASCIZ  /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
10060  066070  052111  041440  042514
10061  066076  051101  042105  041040
10062  066104  020131  040515  051523
10063  066112  052502  020123  047111
10064  066120  052111  020054  051104
10065  066126  053111  020105  047111
10066  066134  047040  052505  051124
10067  066142  046101     000
10068
10069  066145     124  042510  040440  EM52:   .ASCIZ  /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
10070  066152  052124  020116  044502
10071  066160  020124  051511  051440
10072  066166  052105  040440  052106
10073  066174  051105  052040  046511
```

```
10074  066202  047505  052125  053440
10075  066210  052111  020110  047516
10076  066216  051040  050505  042525
10077  066224  052123  023040  023440
10078  066232  051105  023522  051440
10079  066240  052105     000
10080
10081  066243     103  047101  052047  EM53:   .ASCIZ  /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
10082  066250  051040  040505  020104
10083  066256  044124  020105  052101
10084  066264  047124  041040  052111
10085  066272  043040  047522  020115
10086  066300  044124  020105  047447
10087  066306  050120  051517  052111
10088  066314  023505  050040  051117
10089  066322  000124
10090
10091  066324  042522  042514  051501  EM54:   .ASCIZ  /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
10092  066332  020105  047503  046515
10093  066340  047101  020104  042522
10094  066346  047503  047107  055111
10095  066354  042105  053440  042510
10096  066362  020116  051511  052523
10097  066370  042105  041040  020131
10098  066376  047516  026516  042523
10099  066404  055111  047111  020107
10100  066412  047520  052122     000
10101
10102  066417     124  046511  047505  EM55:   .ASCIZ  /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
10103  066424  052125  047440  042516
10104  066432  051455  047510  020124
10105  066440  051511  046040  051505
10106  066446  020123  044124  047101
10107  066454  032440  030060  046440
10108  066462  000123
10109
10110
10111
10112
10113  066464  042524  052123  021440  DH1:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
10114  066472  020040  051105  020122
10115  066500  041520  020040  047520
10116  066506  052122  021440  020040
10117  066514  042522  020107  042101
10118  066522  020122  047503  052116
10119  066530  047105  051524     000
10120  066535     124  051505  020124  DH3:    .ASCIZ  /TEST #  ERR PC  REG ADR PORT A  PORT B/
10121  066542  020043  042440  051122
10122  066550  050040  020103  051040
10123  066556  043505  040440  051104
10124  066564  050040  051117  020124
10125  066572  020101  050040  051117
10126  066600  020124  000102
10127  066604  020040  020040  020040  DH4:    .ASCII  /                SEIZE   ERROR/<15><12>
10128  066612  020040  020040  020040
10129  066620  020040  020040  042523
```

```
10130  066626  055111  020105  020040
10131  066634  051105  047522  006522
10132  066642     012
10133  066643     124  051505  020124          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
10134  066650  020043  042440  051122
10135  066656  050040  020103  050040
10136  066664  051117  020124  020043
10137  066672  050040  051117  020124
10138  066700  020043  051040  043505
10139  066706  040440  051104  043440
10140  066714  047517  020104  020040
10141  066722  041040  042101     000
10142  066727     124  051505  020124  DH5:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR GOOD    BAD/
10143  066734  020043  042440  051122
10144  066742  050040  020103  050040
10145  066750  051117  020124  020043
10146  066756  051040  043505  040440
10147  066764  051104  043440  047517
10148  066772  020104  020040  041040
10149  067000  042101     000
10150  067003     040  020040  020040  DH7:    .ASCII  /                RELSNG  ERROR/<15><12>
10151  067010  020040  020040  020040
10152  067016  020040  020040  051040
10153  067024  046105  047123  020107
10154  067032  042440  051122  051117
10155  067040  005015
10156  067042  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
10157  067050  020040  051105  020122
10158  067056  041520  020040  047520
10159  067064  052122  021440  020040
10160  067072  047520  052122  021440
10161  067100  020040  042522  020107
10162  067106  042101  020122  047507
10163  067114  042117  020040  020040
10164  067122  040502  000104
10165  067126  042524  052123  021440  DH11:   .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
10166  067134  020040  051105  020122
10167  067142  041520  020040  047520
10168  067150  052122  021440  020040
10169  067156  042522  020107  042101
10170  067164  020122  047503  052116
10171  067172  047105  051524     000
10172  067177     040  020040  020040  DH13:   .ASCII  /                SEIZE   ERROR/<15><12>
10173  067204  020040  020040  020040
10174  067212  020040  020040  051440
10175  067220  044505  042532  020040
10176  067226  042440  051122  051117
10177  067234  005015
10178  067236  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR CONTENTS/
10179  067244  020040  051105  020122
10180  067252  041520  020040  047520
10181  067260  052122  021440  020040
10182  067266  047520  052122  021440
10183  067274  020040  042522  020107
10184  067302  042101  020122  047503
10185  067310  052116  047105  051524
```

```
10186  067316     000
10187  067317     040  020040  020040  DH22:   .ASCII  /               RELSNG  SEIZE/<15><12>
10188  067324  020040  020040  020040
10189  067332  020040  020040  051040
10190  067340  046105  047123  020107
10191  067346  051440  044505  042532
10192  067354  005015
10193  067356  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10194  067364  020040  051105  020122
10195  067372  041520  020040  047520
10196  067400  052122  021440  020040
10197  067406  047520  052122  021440
10198  067414     000
10199  067415     040  020040  020040  DH23:   .ASCII  /               SEIZE/<15><12>
10200  067422  020040  020040  020040
10201  067430  020040  020040  051440
10202  067436  044505  042532  005015
10203  067444  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
10204  067452  020040  051105  020122
10205  067460  041520  020040  047520
10206  067466  052122  021440  020040
10207  067474  042522  020107  042101
10208  067502  020122  047503  052116
10209  067510  047105  051524     000
10210  067515     040  020040  020040  DH26:   .ASCII  /               RELSNG/<15><12>
10211  067522  020040  020040  020040
10212  067530  020040  020040  051040
10213  067536  046105  047123  006507
10214  067544     012
10215  067545     124  051505  020124          .ASCIZ  /TEST #  ERR PC  PORT #/
10216  067552  020043  042440  051122
10217  067560  050040  020103  050040
10218  067566  051117  020124  000043
10219  067574  020040  020040  020040  DH31:   .ASCII  /               RELSNG  RQSTNG/<15><12>
10220  067602  020040  020040  020040
10221  067610  020040  020040  042522
10222  067616  051514  043516  020040
10223  067624  050522  052123  043516
10224  067632  005015
10225  067634  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10226  067642  020040  051105  020122
10227  067650  041520  020040  047520
10228  067656  052122  021440  020040
10229  067664  047520  052122  021440
10230  067672     000
10231  067673     124  051505  020124  DH36:   .ASCIZ  /TEST #  ERR PC  PORT #/
10232  067700  020043  042440  051122
10233  067706  050040  020103  050040
10234  067714  051117  020124  000043
10235  067722  042524  052123  021440  DH42:   .ASCIZ  /TEST #  ERR PC/
10236  067730  020040  051105  020122
10237  067736  041520     000
10238  067741     040  020040  020040  DH44:   .ASCII  /               RELSNG  ERROR/<15><12>
10239  067746  020040  020040  020040
10240  067754  020040  020040  051040
10241  067762  046105  047123  020107
```

```
10242  067770  042440  051122  051117
10243  067776  005015
10244  070000  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10245  070006  020040  051105  020122
10246  070014  041520  020040  047520
10247  070022  052122  021440  020040
10248  070030  047520  052122  021440
10249  070036     000
10250  070037     040  020040  020040  DH46:   .ASCII  /               PORT A  PORT B/<15><12>
10251  070044  020040  020040  020040
10252  070052  020040  020040  050040
10253  070060  051117  020124  020101
10254  070066  050040  051117  020124
10255  070074  006502     012
10256  070077     124  051505  020124          .ASCIZ  /TEST #  ERR PC  RHDS1   RHDS1/
10257  070104  020043  042440  051122
10258  070112  050040  020103  051040
10259  070120  042110  030523  020040
10260  070126  051040  042110  030523
10261  070134     000
10262  070135     124  051505  020124  DH55:   .ASCIZ  /TEST #  ERR PC  PORT #  TIMEOUT VALUE (IN MS)/
10263  070142  020043  042440  051122
10264  070150  050040  020103  050040
10265  070156  051117  020124  020043
10266  070164  052040  046511  047505
10267  070172  052125  053040  046101
10268  070200  042525  024040  047111
10269  070206  046440  024523     000
10270
10271          070214                  .EVEN
10272
10273  070214  001234  001116  001226  DT1:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$BDDAT,0
10274  070222  001122  001126  000000
10275  070230  001234  001116  001122  DT3:    .WORD   TSTNUM,$ERRPC,$BDADR,$GDDAT,$BDDAT,0
10276  070236  001124  001126  000000
10277  070244  001234  001116  001226  DT5:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10278  070252  001122  001124  001126
10279  070260  000000
10280  070262  001234  001116  001232  DT6:    .WORD   TSTNUM,$ERRPC,OPPRT,$BDADR,$BDDAT,0
10281  070270  001122  001126  000000
10282  070276  001234  001116  001230  DT7:    .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10283  070304  001226  001122  001124
10284  070312  001126  000000
10285  070316  001234  001116  001230  DT13:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$BDDAT,0
10286  070324  001226  001122  001126
10287  070332  000000
10288  070334  001234  001116  001230  DT22:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,0
10289  070342  001226  000000
10290  070346  001234  001116  001230  DT23:   .WORD   TSTNUM,$ERRPC,SEIZPT,$BDADR,$BDDAT,0
10291  070354  001122  001126  000000
10292  070362  001234  001116  001230  DT31:   .WORD   TSTNUM,$ERRPC,SEIZPT,OPPRT,0
10293  070370  001232  000000
10294  070374  001234  001116  001230  DT36:   .WORD   TSTNUM,$ERRPC,SEIZPT,0
10295  070402  000000
10296  070404  001234  001116  001226  DT37:   .WORD   TSTNUM,$ERRPC,PTNBR,0
10297  070412  000000
```

```
10298  070414  001234  001116  000000  DT42:   .WORD   TSTNUM,$ERRPC,0
10299  070422  001234  001116  001162  DT46:   .WORD   TSTNUM,$ERRPC,$TMP2,$TMP3,0
10300  070430  001164  000000
10301  070434  001234  001116  001232  DT54:   .WORD   TSTNUM,$ERRPC,OPPRT,SEIZPT,0
10302  070442  001230  000000
10303  070446  001234  001116  001230  DT55:   .WORD   TSTNUM,$ERRPC,SEIZPT,TIME,0
10304  070454  001244  000000
10305
10306  070460     000     000     000  DF1:    .BYTE   0,0,0,0,0
10307  070463     000     000
10308  070465     000     000     000  DF5:    .BYTE   0,0,0,0,0,0
10309  070470     000     000     000
10310  070473     000     000     000  DF7:    .BYTE   0,0,0,0,0,0,0
10311  070476     000     000     000
10312  070501     000
10313  070502     000     000     000  DF31:   .BYTE   0,0,0,0
10314  070505     000
10315  070506     000     000     000  DF36:   .BYTE   0,0,0
10316  070511     000     000          DF42:   .BYTE   0,0
10317  070513     000     000     000  DF55:   .BYTE   0,0,0,1
10318  070516     001
10319
10320          070520                       .EVEN
10321
10322
10323                          ;;*****************************************************************************
10324
10325                          .SBTTL  CONSTANTS, TABLES, ETC
10326
10327                          ;;*****************************************************************************
10328
10329                          ;TABLE OF TEST STARTING ADDRESSES
10330
10331  070520  002712          TSTADR: .WORD   TST1+2          ;STARTING ADDRESS OF TEST 1
10332  070522  004244                  .WORD   TST2+2          ;STARTING ADDRESS OF TEST 2
10333  070524  005670                  .WORD   TST3+2          ;STARTING ADDRESS OF TEST 3
10334  070526  007314                  .WORD   TST4+2          ;STARTING ADDRESS OF TEST 4
10335  070530  010500                  .WORD   TST5+2          ;STARTING ADDRESS OF TEST 5
10336  070532  011664                  .WORD   TST6+2          ;STARTING ADDRESS OF TEST 6
10337  070534  012702                  .WORD   TST7+2          ;STARTING ADDRESS OF TEST 7
10338  070536  013720                  .WORD   TST10+2         ;STARTING ADDRESS OF TEST 10
10339  070540  014370                  .WORD   TST11+2         ;STARTING ADDRESS OF TEST 11
10340  070542  015040                  .WORD   TST12+2         ;STARTING ADDRESS OF TEST 12
10341  070544  016442                  .WORD   TST13+2         ;STARTING ADDRESS OF TEST 13
10342  070546  020044                  .WORD   TST14+2         ;STARTING ADDRESS OF TEST 14
10343  070550  021374                  .WORD   TST15+2         ;STARTING ADDRESS OF TEST 15
10344  070552  022724                  .WORD   TST16+2         ;STARTING ADDRESS OF TEST 16
10345  070554  024212                  .WORD   TST17+2         ;STARTING ADDRESS OF TEST 17
10346  070556  025124                  .WORD   TST20+2         ;STARTING ADDRESS OF TEST 20
10347  070560  026036                  .WORD   TST21+2         ;STARTING ADDRESS OF TEST 21
10348  070562  027242                  .WORD   TST22+2         ;STARTING ADDRESS OF TEST 22
10349  070564  030446                  .WORD   TST23+2         ;STARTING ADDRESS OF TEST 23
10350  070566  031502                  .WORD   TST24+2         ;STARTING ADDRESS OF TEST 24
10351  070570  032536                  .WORD   TST25+2         ;STARTING ADDRESS OF TEST 25
10352  070572  032536                  .WORD   TST25+2         ;STARTING ADDRESS OF TEST 25
10353  070574  033752                  .WORD   TST26+2         ;STARTING ADDRESS OF TEST 26
```

```
10354  070576  035166                          .WORD   TST27+2         ;STARTING ADDRESS OF TEST 27
10355  070600  037204                          .WORD   TST30+2         ;STARTING ADDRESS OF TEST 30
10356  070602  037726                          .WORD   TST31+2         ;STARTING ADDRESS OF TEST 31
10357  070604  041214                          .WORD   TST32+2         ;STARTING ADDRESS OF TEST 32
10358  070606  042502                          .WORD   TST33+2         ;STARTING ADDRESS OF TEST 33
10359  070610  043620                          .WORD   TST34+2         ;STARTING ADDRESS OF TEST 34
10360  070612  044736                          .WORD   TST35+2         ;STARTING ADDRESS OF TEST 35
10361  070614  045600                          .WORD   TST36+2         ;STARTING ADDRESS OF TEST 36
10362  070616  046442                          .WORD   TST37+2         ;STARTING ADDRESS OF TEST 37
10363  070620  047436                          .WORD   TST40+2         ;STARTING ADDRESS OF TEST 40
10364  070622  050432                          .WORD   TST41+2         ;STARTING ADDRESS OF TEST 41
10365  070624  051442                          .WORD   TST42+2         ;STARTING ADDRESS OF TEST 42
10366  070626  052452                          .WORD   TST43+2         ;STARTING ADDRESS OF TEST 43
10367  070630  054040                          .WORD   TST44+2         ;STARTING ADDRESS OF TEST 44
10368  070632  055426                          .WORD   TST45+2         ;STARTING ADDRESS OF TEST 45
10369  070634  056344                          .WORD   TST46+2         ;STARTING ADDRESS OF TEST 46
10370
10371                                  ;ATTENTION BIT TABLE
10372
10373  070636     001                  ATABIT: .BYTE   1               ;ATTENTION BIT FOR DRIVE 0
10374  070637     002                          .BYTE   2               ;ATTENTION BIT FOR DRIVE 1
10375  070640     004                          .BYTE   4               ;ATTENTION BIT FOR DRIVE 2
10376  070641     010                          .BYTE   10              ;ATTENTION BIT FOR DRIVE 3
10377  070642     020                          .BYTE   20              ;ATTENTION BIT FOR DRIVE 4
10378  070643     040                          .BYTE   40              ;ATTENTION BIT FOR DRIVE 5
10379  070644     100                          .BYTE   100             ;ATTENTION BIT FOR DRIVE 6
10380  070645     200                          .BYTE   200             ;ATTENTION BIT FOR DRIVE 7
10381
10382  070646  000046                  MAXTN:  .WORD   $TN-1            ;MAXIMUM TEST NUMBER
10383
10384          000001                          .END
```

DERPPA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ACL    = 000040          1393#
ACU    = 100000          1363#
ADDRIS   062267          1974    9672#
ADRERR   062042          1910    9645#
AOE    = 001000          1293#
ASR1     001224          1511#   1929*   6124    6412    8352    8571    8769    8770    8895    8896
ATA    = 100000          1280#   2334    2338    2359    2365    2584    2588    2609    2615    2853    2857    2878    2884
                         3048    3052    3073    3079    3159    3163    3184    3190    3206    3210    3223    3227    3320
                         3324    3345    3351    3367    3371    3384    3388    3468    3472    3493    3499    3562    3566
                         3587    3593    3778    3782    3805    3811    3996    4000    4023    4029    4108    4110    4114
                         4128    4130    4134    4167    4171    4194    4198    4221    4227    4242    4244    4248    4325
                         4327    4331    4345    4347    4351    4384    4388    4411    4415    4438    4444    4459    4461
                         4465    4540    4542    4546    4560    4562    4566    4586    4590    4639    4641    4645    4656
                         4658    4662    4768    4772    4905    4909    5031    5057    5061    5072    5074    5078    5101
                         5105    5226    5252    5256    5267    5269    5273    5296    5300    5460    5464    5623    5627
                         5789    5823    5827    5988    6022    6026    6179    6213    6217    6284    6318    6322    6429
                         6433    6593    6595    6599    6617    6621    6644    6650    6667    6671    6811    6813    6817
                         6835    6839    6862    6868    6885    6889    7051    7055    7236    7240    7381    7385    7533
                         7537    7722    7726    7903    7907    8020    8022    8026    8043    8047    8070    8074    8183
                         8185    8189    8206    8210    8233    8237    8371    8405    8409    8590    8624    8628    8807
                         8811    8834    8840    8933    8937    8960    8966
ATABIT   070636          1929   10373#
ATO    = 000001          1313#
AT1    = 000002          1314#
AT2    = 000004          1315#
AT3    = 000010          1316#
AT4    = 000020          1317#
AT5    = 000040          1318#
AT6    = 000100          1319#
AT7    = 000200          1320#
A16    = 000400          1255#
A17    = 001000          1256#
BADNO    062227          1957    9666#
BAI    = 000010          1226#
BITO   = 000001          1193#
BIT00  = 000001          1183#    1193
BIT01  = 000002          1182#    1192
BIT02  = 000004          1181#    1191
BIT03  = 000010          1180#    1190
BIT04  = 000020          1179#    1189
BIT05  = 000040          1178#    1188
BIT06  = 000100          1177#    1187
BIT07  = 000200          1176#    1186
BIT08  = 000400          1175#    1185
BIT09  = 001000          1174#    1184
BIT1   = 000002          1192#
BIT10  = 002000          1173#    9145
BIT11  = 004000          1172#    9111
BIT12  = 010000          1171#
BIT13  = 020000          1170#    9152
BIT14  = 040000          1169#
BIT15  = 100000          1168#
BIT2   = 000004          1191#
BIT3   = 000010          1190#
BIT4   = 000020          1189#
BIT5   = 000040          1188#
```

```
BIT6   = 000100          1187#
BIT7   = 000200          1186#
BIT8   = 000400          1185#
BIT9   = 001000          1184#
BPTVEC= 000014           1200#
CHANGE  002560           1448    1969#   1991
CKCLK   057264           1930    1943    8983#
CKCLK1  057334           8983    8992#
CKCLK2  057376           8993    9000#
CKCLK3  057406           8991    8999    9002#
CKERR   001236           1516#   2040*   2053*   2055    2061*   2074*   2076    2085*   2093*   2097*   2105*   2113*   2126*
                         2128*   2141*   2145*   2158*   2160*   2173*   2759*   2767*   2771*   2784*   2786*   2799*   2811*
                         2824*   2954*   2962*   2966*   2979*   2981*   2994*   3006*   3019*   3200*   3213*   3217*   3230*
                         3361*   3374*   3378*   3391*   3687*   3700*   3704*   3717*   3730*   3743*   3747*   3760*   3905*
                         3918*   3922*   3935*   3948*   3961*   3965*   3978*   4104*   4117*   4119    4124*   4137*   4139
                         4161*   4174*   4238*   4251*   4321*   4334*   4336    4341*   4354*   4356    4378*   4391*   4455*
                         4468*   4536*   4549*   4551    4556*   4569*   4571    4635*   4648*   4652*   4665*   5051*   5064*
                         5068*   5081*   5246*   5259*   5263*   5276*   5425*   5438*   5588*   5601*   5760*   5768*   5774
                         5959*   5967*   5973    6152*   6165*   6257*   6270*   6556*   6569*   6575*   6583*   6589*   6602*
                         6661*   6674*   6774*   6787*   6793*   6801*   6807*   6820*   6879*   6892*   6976*   6989*   6991
                         7006*   7014*   7018*   7026*   7161*   7174*   7176    7191*   7199*   7203*   7211*   7671*   7679*
                         7681    7691*   7699*   7701    7852*   7860*   7862    7872*   7880*   7882    8016*   8029*   8037*
                         8050*   8179*   8192*   8200*   8213*   8456*   8464*   8470*   8478*   8484*   8492*   8675*   8683*
                         8689*   8697*   8703*   8711*
CLOCK   057416           8987    8996    9007#
CLR    = 000040          1228#   2057    2078    3723    3941    4155    4372    4628    6994    7179
CSF    = 000002          1350#
CSU    = 000010          1352#
DCK    = 100000          1299#
DCL    = 000100          1394#
DE1    = 000040          1270#
DFF20  = 000002          1266#
DF1     070460           1562    1569    1576    1618    1688    1695    1781    1788    1816    1830    1858    10306#
DF31    070502           1681    1711    1732    1809    1823    1865    10313#
DF36    070506           1625    1667    1674    1703    1725    1746    1753    1760    1767    1774    10315#
DF42    070511           1795    1802    10316#
DF5     070465           1590    1597    1611    1632    1639    1739    1837    1844    1851    10308#
DF55    070513           1872    10317#
DF7     070473           1583    1604    1646    1653    1660    1718    10310#
DH1     066464           1560    1567    1779    1814    10113#
DH11    067126           1616    10165#
DH13    067177           1595    1630    1637    1835    1849    10172#
DH22    067317           1679    1709    1863    10187#
DH23    067415           1686    1693    1786    1828    1856    10199#
DH26    067515           10210#
DH3     066535           1574    10120#
DH31    067574           1730    10219#
DH36    067673           1623    1665    1672    1701    1723    1744    1751    1758    1765    1772    10231#
DH4     066604           1581    10127#
DH42    067722           1793    1800    10235#
DH44    067741           1807    10238#
DH46    070037           1821    10250#
DH5     066727           1588    1609    1737    1842    10142#
DH55    070135           1870    10262#
DH7     067003           1602    1644    1651    1658    1716    10150#
DIGB   = 000004          1267#
```

DERPPA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DISPLA= 177570          1115#   9122*   9144*
DLT    = 100000         1238#
DL64   = 000020         1269#
DMD    = 000001         1303#
DPR    = 000400         1273#   2132    2164    2269    2330    2519    2580    2775    2815    2849    2970    3010    3044
                        3155    3316    3464    3558    3668    3708    3751    3774    3886    3926    3969    3992    4190
                        4407    4742    4764    4879    4901    5031    5097    5226    5292    5411    5456    5574    5619
                        5741    5789    5819    5940    5988    6018    6156    6179    6209    6261    6284    6314    6425
                        7047    7232    7377    7529    7718    7899    8066    8229    8371    8401    8590    8620    8803
                        8929
DRY    = 000200         1272#   2132    2164    2269    2330    2519    2580    2775    2815    2849    2970    3010    3044
                        3155    3316    3464    3558    3668    3708    3751    3774    3886    3926    3969    3992    4190
                        4407    4742    4764    4879    4901    5031    5097    5226    5292    5411    5456    5574    5619
                        5741    5789    5819    5940    5988    6018    6156    6179    6209    6261    6284    6314    6425
                        7047    7232    7377    7529    7718    7899    8010    8066    8173    8229    8371    8401    8590
                        8620    8803    8929
DTE    = 010000         1296#
DTSY   = 000200         1309#
DT1      070214         1561    1568    1617    1815    1857    10273#
DT13     070316         1596    1631    1638    1836    1850    10285#
DT22     070334         1680    1710    10288#
DT23     070346         1687    1694    1780    1787    1829    10290#
DT3      070230         1575    10275#
DT31     070362         1731    1808    10292#
DT36     070374         1702    1724    1745    1752    1766    10294#
DT37     070404         1624    1666    1673    1759    1773    10296#
DT42     070414         1794    1801    10298#
DT46     070422         1822    10299#
DT5      070244         1589    1610    1738    1843    10277#
DT54     070434         1864    10301#
DT55     070446         1871    10303#
DT6      070262         10280#
DT7      070276         1582    1603    1645    1652    1659    1717    10282#
DVA    = 004000         1258#
ECH    = 000100         1290#
ECI    = 004000         1375#
EMTVEC= 000030          1203#   1891*   1892*
EM1      062471         1559    9703#
EM10     063076         1608    9757#
EM11     063126         1615    9762#
EM12     063212         1622    9772#
EM13     063262         1629    9780#
EM14     063347         1636    9790#
EM15     063412         1643    9797#
EM16     063471         1650    9806#
EM17     063544         1657    9815#
EM2      062512         1566    9707#
EM20     063623         1664    9824#
EM21     063703         1671    9834#
EM22     063756         1678    9843#
EM23     064043         1685    9853#
EM24     064111         1692    9861#
EM25     064170         1700    9870#
EM26     064233         1708    9877#
EM27     064320         1715    9887#
EM3      062534         1573    9711#
```

```
EM30    064376          1722    9896#
EM31    064473          1729    9908#
EM32    064550          1736    9917#
EM33    064621          1743    9925#
EM34    064723          1750    9938#
EM35    065026          1757    9951#
EM36    065105          1764    9960#
EM37    065156          1771    9968#
EM4     062616          1580    9721#
EM40    065224          1778    9976#
EM41    065301          1785    9985#
EM42    065343          1792    9992#
EM43    065431          1799   10003#
EM44    065506          1806   10012#
EM45    065563          1813   10021#
EM46    065642          1820   10030#
EM47    065711          1827   10038#
EM5     062647          1587    9727#
EM50    065777          1834   10049#
EM51    066062          1841   10059#
EM52    066145          1848   10069#
EM53    066243          1855   10081#
EM54    066324          1862   10091#
EM55    066417          1869   10102#
EM6     062715          1594    9735#
EM7     063015          1601    9747#
ENTERA  062012          1905    9641#
ERR     = 040000        1279#   6560    6778
ERRVEC= 000004          1196#   8983*   8984*   8993*   9002*   9098    9099*   9101*   9104*
EXEC    002410          1445    1931    1938#   1958    2022    2225    2476    2727    2922    3114    3275    3434    3529
                        3634    3852    4071    4288    4503    4697    4833    4978    5173    5366    5529    5696    5895
                        6091    6384    6496    6714    6939    7124    7306    7458    7616    7797    7975    8138    8311
                        8530    8738    8864    9048
EXT1    = 000001        1330#
EXT10   = 000010        1333#
EXT2    = 000002        1331#
EXT20   = 000020        1334#
EXT4    = 000004        1332#
EXT40   = 000040        1335#
FEN     = 000200        1356#
FER     = 000020        1288#
FMT22   = 010000        1376#   2806    3001
GNS     = ****** U      1438    9613    9614    9615    9616    9617    9618    9619    9620    9621    9622
GO      = 000001        1252#
GRV     = 000010        1268#
HCE     = 000200        1291#
HCI     = 002000        1374#
HCRC    = 000400        1292#
IAE     = 002000        1294#
IE      = 000100        1253#
ILF     = 000001        1284#
ILR     = 000002        1285#
IOTVEC= 000020          1201#   1889*   1890*
IR      = 000100        1229#
IXE     = 004000        1360#
KYBCTL  001266          1528#   1945*   1962*   2019    2023*   2222    2226*   2473    2477*   2724    2728*   2919    2923*
```

```
                        3111    3115*   3272    3276*   3431    3435*   3526    3530*   3631    3635*   3849    3853*   4068
                        4072*   4285    4289*   4500    4504*   4694    4698*   4830    4834*   4975    4979*   5170    5174*
                        5363    5367*   5526    5530*   5693    5697*   5892    5896*   6088    6092*   6381    6385*   6493
                        6497*   6711    6715*   6936    6940*   7121    7125*   7303    7307*   7455    7459*   7613    7617*
                        7794    7798*   7972    7976*   8135    8139*   8308    8312*   8527    8531*   8735    8739*   8861
                        8865*   9046
LST     = 002000        1275#
MAXTN   = 070646        1955    10382#
MCLK    = 000002        1304#
MCPE    = 020000        1259#   5770    5969    8466    8480    8494    8685    8699    8713
MHS     = 001000        1358#
MINX    = 000004        1305#
MOL     = 010000        1277#   2132    2164    2269    2330    2519    2580    2775    2815    2849    2970    3010    3044
                        3155    3316    3464    3558    3668    3708    3751    3774    3886    3926    3969    3992    4190
                        4407    4742    4764    4879    4901    5031    5097    5226    5292    5411    5456    5574    5619
                        5741    5789    5819    5940    5988    6018    6156    6179    6209    6261    6284    6314    6425
                        7047    7232    7377    7529    7718    7899    8066    8229    8371    8401    8590    8620    8803
                        8929
MPE     = 000400        1231#
MRD     = 000020        1307#
MSE     = 000020        1353#
MSTCK   = 000010        1306#
MWR     = 000040        1308#
MXF     = 001000        1232#
NED     = 010000        1235#   2046    2050    2067    2071
NEM     = 004000        1234#
NHS     = 002000        1359#
NOATA   = 000001        1876#   2359    2365    2609    2615    2878    2884    3073    3079    3184    3190    3345    3351
                        3493    3499    3587    3593    3805    3811    4023    4029    4221    4227    4438    4444    4482#
                        4613    4618    4795    4800    4932    4937    5018#   5084#   5128    5133    5213#   5279#   5323
                        5328    5487    5492    5650    5655    5850    5855    6049    6054    6240    6245    6345    6350
                        6456    6461    6607#   6644    6650    6655#   6825#   6862    6868    6873#   7078    7083    7263
                        7268    7408    7413    7560    7565    7749    7754    7930    7935    8097    8102    8260    8265
                        8432    8437    8651    8656    8791#   8834    8840    8845#   8917#   8960    8966    8971#
NOCLOC  062140          1932    9656#
NORESP  062401          1988    9686#
NOSEIZ  001240          1517#
NTRH11  062347          1978    9681#
OCYL    = 100000        1396#
OFREV   = 000200        1373#
OF100   = 000004        1369#
OF200   = 000010        1370#
OF25    = 000001        1367#
OF400   = 000020        1371#
OF50    = 000002        1368#
OF800   = 000040        1372#
OPI     = 020000        1297#
OPPRT   001232          1514#   2257*   2507*   3140*   3301*   3656*   3874*   4150*   4367*   4730*   4867*   5009*   5204*
                        5399*   5562*   5729*   5928*   6168*   6273*   6526*   6744*   6969*   7154*   7337*   7489*   7647*
                        7828*   8340*   8559*   10280   10292   10301
OR      = 000200        1230#
PAR     = 000010        1287#
PAT     = 000020        1227#
PC      =%000007        1127#   1930*   1943*   8346*   8360*   8498*   8565*   8579*   8717*   9003*   9053*   9056*   9070*
                        9154*   9215*   9252*   9259*   9264*
PGE     = 002000        1233#
```

```
PGM    = 001000          1274#  2117   2119   2123   2149   2151   2155   2269   2330   2519   2580   2775   2815
                         2849   2970   3010   3044   3155   3316   3464   3558   3668   3708   3751   3774   3886
                         3926   3969   3992   4190   4407   4742   4764   4879   4901   5031   5097   5226   5292
                         5411   5456   5574   5619   5741   5789   5819   5940   5988   6018   6156   6179   6209
                         6261   6284   6314   6425   7047   7232   7377   7529   7718   7899   8066   8229   8371
                         8401   8590   8620   8803   8929
FIP    = 020000          1278#
PIRQ   = 177772          1113#
PIRQVE= 000240           1207#
PLU    = 020000          1362#
PORTA   001216           1508#  1907*  1908   1912   1915   1919   1925   1928   2037   2038   2083   2084   2111
                         2112   2179   2252   2253   2266   2267   2331   2350   2352   2358   2505   2506   2507
                         2531   2532   2581   2600   2602   2608   2625   2739   2740   2747   2769   2770   2850
                         2869   2871   2877   2952   2953   3021   3022   3045   3064   3066   3072   3137   3138
                         3146   3147   3156   3175   3177   3183   3198   3199   3301   3317   3336   3338   3344
                         3376   3377   3400   3401   3446   3447   3448   3465   3484   3486   3492   3559   3578
                         3580   3586   3651   3652   3665   3666   3702   3703   3745   3746   3765   3766   3775
                         3794   3796   3804   3872   3873   3874   3903   3904   3946   3947   3993   4012   4014
                         4022   4087   4095   4102   4103   4147   4148   4156   4181   4182   4191   4210   4212
                         4220   4304   4312   4339   4340   4367   4408   4427   4429   4437   4453   4454   4519
                         4527   4534   4535   4583   4602   4604   4612   4633   4634   4712   4725   4726   4739
                         4740   4755   4756   4765   4784   4786   4794   4849   4865   4866   4867   4902   4921
                         4923   4931   4993   5009   5010   5011   5034   5035   5045   5066   5067   5088   5089
                         5098   5117   5119   5127   5188   5201   5202   5219   5220   5232   5233   5244   5245
                         5293   5312   5314   5322   5381   5397   5398   5399   5419   5420   5457   5476   5478
                         5486   5544   5557   5558   5571   5572   5610   5611   5620   5639   5641   5649   5711
                         5727   5728   5729   5753   5754   5792   5793   5803   5810   5811   5820   5839   5841
                         5849   5910   5923   5924   5937   5938   5981   5982   5994   5995   6019   6038   6040
                         6048   6106   6119   6125   6126   6168   6182   6183   6193   6200   6201   6210   6229
                         6231   6239   6255   6256   6272   6277   6278   6290   6291   6315   6334   6336   6344
                         6399   6426   6445   6447   6455   6511   6523   6524   6550   6551   6614   6633   6635
                         6643   6679   6729   6744   6756   6757   6832   6851   6853   6861   6877   6878   6954
                         6966   6967   6995   6996   7016   7017   7038   7039   7048   7067   7069   7077   7139
                         7154   7189   7190   7233   7252   7254   7262   7321   7334   7335   7378   7397   7399
                         7407   7473   7489   7513   7514   7530   7549   7551   7559   7631   7647   7648   7649
                         7690   7709   7710   7719   7738   7740   7748   7812   7825   7826   7836   7837   7900
                         7919   7921   7929   7990   7998   7999   8000   8057   8058   8067   8086   8088   8096
                         8153   8194   8195   8230   8249   8251   8259   8326   8337   8338   8347   8348   8358
                         8359   8364   8365   8377   8378   8402   8421   8423   8431   8545   8559   8563   8564
                         8572   8573   8593   8594   8604   8611   8612   8621   8640   8642   8650   8753   8761
                         8804   8823   8825   8833   8879   8890   8899   8900   8901   8920   8921   8930   8949
                         8951   8959
PORTAI  062064           1918   9648#
PORTB   001220           1509#  1912*  1913*  1914*  1917*  1922   2058   2059   2095   2096   2143   2144   2181
                         2255   2256   2257   2281   2282   2335   2346   2347   2364   2375   2502   2503   2516
                         2517   2585   2596   2597   2614   2757   2758   2826   2827   2854   2865   2866   2883
                         2934   2935   2942   2964   2965   3049   3060   3061   3078   3140   3160   3171   3172
                         3189   3215   3216   3239   3240   3298   3299   3307   3308   3321   3332   3333   3350
                         3359   3360   3469   3480   3481   3498   3540   3541   3542   3563   3574   3575   3592
                         3654   3655   3656   3685   3686   3728   3729   3779   3790   3791   3810   3869   3870
                         3883   3884   3920   3921   3963   3964   3983   3984   3997   4008   4009   4028   4090
                         4122   4123   4150   4195   4206   4207   4226   4236   4237   4307   4319   4320   4364
                         4365   4373   4398   4399   4412   4423   4424   4443   4522   4554   4555   4587   4598
                         4599   4617   4650   4651   4716   4728   4729   4730   4769   4780   4781   4799   4853
                         4862   4863   4876   4877   4892   4893   4906   4917   4918   4936   4997   5006   5007
                         5024   5025   5037   5038   5049   5050   5102   5113   5114   5132   5192   5204   5205
```

```
                                5206    5229    5230    5240    5261    5262    5283    5284    5297    5308    5309    5327    5385
                                5394    5395    5408    5409    5447    5448    5461    5472    5473    5491    5548    5560    5561
                                5562    5582    5583    5624    5635    5636    5654    5715    5724    5725    5738    5739    5782
                                5783    5795    5796    5824    5835    5836    5854    5914    5926    5927    5928    5952    5953
                                5991    5992    6002    6009    6010    6023    6034    6035    6053    6110    6133    6134    6140
                                6141    6167    6172    6173    6185    6186    6214    6225    6226    6244    6273    6287    6288
                                6298    6305    6306    6319    6330    6331    6349    6403    6430    6441    6442    6460    6515
                                6526    6538    6539    6618    6629    6630    6649    6659    6660    6733    6741    6742    6768
                                6769    6836    6847    6848    6867    6897    6958    6969    7004    7005    7052    7063    7064
                                7082    7143    7151    7152    7180    7181    7201    7202    7223    7224    7237    7248    7249
                                7267    7325    7337    7361    7362    7382    7393    7394    7412    7477    7486    7487    7534
                                7545    7546    7564    7635    7644    7645    7655    7656    7723    7734    7735    7753    7816
                                7828    7829    7830    7871    7890    7891    7904    7915    7916    7934    7994    8031    8032
                                8071    8082    8083    8101    8157    8161    8162    8163    8220    8221    8234    8245    8246
                                8264    8330    8340    8344    8345    8353    8354    8374    8375    8385    8392    8393    8406
                                8417    8418    8436    8549    8556    8557    8566    8567    8577    8578    8583    8584    8596
                                8597    8625    8636    8637    8655    8757    8764    8773    8774    8775    8794    8795    8808
                                8819    8820    8839    8883    8887    8934    8945    8946    8965
PORTBI  062112                  1921    9652#
PORTC   001222                  1510*   1925*   1926*   1927*   6118    6407
PRE     = 000020                1392#
PRO     = 000000                1130#
PR1     = 000040                1131#
PR2     = 000100                1132#
PR3     = 000140                1133#
PR4     = 000200                1134#
PR5     = 000240                1135#
PR7     = 000340                1137#
PS      = 177776                1110#   1111    1883*   1939*   1970*   2237*   2306*   2323*   2487*   2556*   2573*
PSEL    = 002000                1257#
PSU     = 000001                1389#
PSW     = 177776                1111#
PTNBR   001226                  1512#   2038*   2059*   2084*   2096*   2112*   2144*   2256*   2267*   2282*   2346*   2350*   2358*
                                2364*   2375*   2506*   2517*   2532*   2596*   2600*   2608*   2614*   2625*   2740*   2758*   2770*
                                2827*   2865*   2869*   2877*   2883*   2935*   2953*   2965*   3022*   3060*   3064*   3072*   3078*
                                3147*   3171*   3175*   3183*   3189*   3199*   3216*   3240*   3308*   3332*   3336*   3344*   3350*
                                3360*   3377*   3401*   3447*   3480*   3484*   3492*   3498*   3541*   3574*   3578*   3586*   3592*
                                3655*   3666*   3686*   3703*   3729*   3746*   3766*   3790*   3794*   3804*   3810*   3873*   3884*
                                3904*   3921*   3947*   3964*   3984*   4008*   4012*   4022*   4028*   4103*   4123*   4182*   4206*
                                4210*   4220*   4226*   4237*   4320*   4340*   4399*   4423*   4427*   4437*   4443*   4454*   4535*
                                4555*   4598*   4602*   4612*   4617*   4634*   4651*   4729*   4740*   4756*   4780*   4784*   4794*
                                4799*   4866*   4877*   4893*   4917*   4921*   4931*   4936*   5011*   5025*   5035*   5038*   5045*
                                5050*   5067*   5089*   5113*   5117*   5127*   5132*   5206*   5220*   5230*   5233*   5240*   5245*
                                5262*   5284*   5308*   5312*   5322*   5327*   5398*   5409*   5420*   5448*   5472*   5476*   5486*
                                5491*   5561*   5572*   5583*   5611*   5635*   5639*   5649*   5654*   5728*   5739*   5754*   5783*
                                5793*   5796*   5803*   5811*   5835*   5839*   5849*   5854*   5927*   5938*   5953*   5982*   5992*
                                5995*   6002*   6010*   6034*   6038*   6048*   6053*   6126*   6134*   6141*   6173*   6183*   6186*
                                6193*   6201*   6225*   6229*   6239*   6244*   6256*   6278*   6288*   6291*   6298*   6306*   6330*
                                6334*   6344*   6349*   6441*   6445*   6455*   6460*   6539*   6551*   6629*   6633*   6643*   6649*
                                6660*   6757*   6769*   6847*   6851*   6861*   6867*   6878*   6996*   7005*   7017*   7039*   7063*
                                7067*   7077*   7082*   7181*   7190*   7202*   7224*   7248*   7252*   7262*   7267*   7362*   7393*
                                7397*   7407*   7412*   7514*   7545*   7549*   7559*   7564*   7649*   7656*   7690*   7710*   7734*
                                7738*   7748*   7753*   7830*   7837*   7871*   7891*   7915*   7919*   7929*   7934*   7999*   8032*
                                8058*   8082*   8086*   8096*   8101*   8162*   8195*   8221*   8245*   8249*   8259*   8264*   8345*
                                8348*   8354*   8359*   8365*   8375*   8378*   8385*   8393*   8417*   8421*   8431*   8436*   8564*
                                8567*   8573*   8578*   8584*   8594*   8597*   8604*   8612*   8636*   8640*   8650*   8655*   8774*
```

DERPPA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    8795*  8819*  8823*  8833*  8839*  8900*  8921*  8945*  8949*  8959*  8965*  10273  10277
                   10282  10285  10288  10296
PWRVEC= 000024      1202*
RDCHR = 104412      9440   9618*
RDLIN = 104414      9508   9619*
RDOCT = 104416      1906   1951   1979   9620*
RDY   = 000200      1254*
RELERR  001242      1518*  2327*  2355*  2577*  2605*  2846*  2874*  3041*  3069*  3152*  3180*  3195   3313*
                    3341*  3356   3461*  3489*  3555*  3583*  3771*  3799*  3989*  4017*  4187*  4215*  4404*
                    4432*  4579*  4607*  4622   4761*  4789*  4898*  4926*  5030*  5094*  5122*  5225*  5289*
                    5317*  5453*  5481*  5616*  5644*  5788*  5816*  5844*  5987*  6015*  6043*  6178*  6206*
                    6234*  6283*  6311*  6339*  6422*  6450*  6610*  6638*  6828*  6856*  7044*  7072*  7229*
                    7257*  7374*  7402*  7526*  7554*  7715*  7743*  7896*  7924*  8063*  8091*  8226*  8254*
                    8370*  8398*  8426*  8589*  8617*  8645*  8800*  8828*  8926*  8954*
RELOK = 000001      1876*  2356   2606   2875   3070   3181   3342   3490   3584   3609*  3800   4018   4216
                    4433   4608   4790   4927   5123   5318   5482   5645   5845   6044   6235   6340   6451
                    6639   6857   7073   7258   7403   7555   7744   7925   8092   8255   8427   8646   8829
                    8955
RESREG= 104422      9622*
RESVEC= 000010      1197*
RHAS  = 000016      1417*  6124*  6414*  8352*  8571*  8767   8777   8893   8903
RHBA  = 000004      1412*
RHCA  = 000034      1424*  2288   2412   2538   2662
RHCC  = 000036      1425*  2287   2417   2537   2667
RHCS1 = 000000      1410*  2752*  2787   2788   2805*  2947*  2982   2983   3000*  3148*  3309*  3453*  3547*
                    3680*  3767*  3800*  3801*  3898*  3985*  4018*  4019*  4183*  4216*  4217*  4400*  4433*
                    4434*  4608*  4609*  4714*  4715*  4718*  4719*  4727   4757*  4750*  4791*  4851*  4852*
                    4855*  4856*  4864   4894*  4927*  4928*  4995*  4996*  4999*  5000*  5026*  5090*  5123*
                    5124*  5190*  5191*  5194*  5195*  5221*  5285*  5318*  5319*  5383*  5384*  5387*  5388*
                    5396   5426   5427   5449*  5482*  5483*  5546*  5547*  5550*  5551*  5559   5589   5590
                    5612*  5645*  5646*  5713*  5714*  5717*  5718*  5755*  5769   5773*  5784*  5812*  5845*
                    5846*  5912*  5913*  5916*  5917*  5954*  5968   5972*  5983*  6011*  6044*  6045*  6108*
                    6109*  6112*  6113*  6174*  6202*  6235*  6236*  6279*  6307*  6340*  6341*  6401*  6402*
                    6405*  6406*  6451*  6452*  6513*  6514*  6517*  6518*  6639*  6640*  6681*  6682*  6731*
                    6732*  6735*  6736*  6857*  6858*  6899*  6900*  6956*  6957*  6960*  6961*  6975*  6977
                    6978   6997*  7032*  7040*  7073*  7074*  7141*  7142*  7145*  7146*  7160*  7162   7163
                    7182*  7217*  7225*  7258*  7259*  7323*  7324*  7327*  7328*  7403*  7404*  7475*  7476*
                    7479*  7480*  7555*  7556*  7633*  7634*  7637*  7638*  7661*  7711*  7744*  7745*  7814*
                    7815*  7818*  7819*  7842*  7892*  7925*  7926*  7992*  7993*  7996*  7997*  8005*  8059*
                    8092*  8093*  8155*  8156*  8159*  8160*  8168*  8222*  8255*  8256*  8328*  8329*  8332*
                    8333*  8366*  8394*  8427*  8428*  8465   8469*  8479   8483*  8493   8497*  8547*  8548*
                    8551*  8552*  8585*  8613*  8646*  8647*  8684   8688*  8698   8702*  8712   8716*  8755*
                    8756*  8759*  8760*  8796*  8829*  8830*  8881*  8882*  8885*  8886*  8922*  8955*  8956*
RHCS2 = 000010      1414*  2037*  2041   2042   2057*  2058*  2062   2063   2078*  2083*  2095*  2111*  2143*
                    2179*  2181*  2252*  2255*  2266*  2281*  2331*  2335*  2347*  2352*  2502*  2505*  2516*
                    2531*  2581*  2585*  2597*  2602*  2739*  2757*  2769*  2826*  2850*  2854*  2866*  2871*
                    2934*  2952*  2964*  3021*  3045*  3049*  3061*  3066*  3137*  3146*  3156*  3160*  3172*
                    3177*  3198*  3215*  3239*  3298*  3307*  3317*  3321*  3333*  3338*  3359*  3376*  3400*
                    3446*  3465*  3469*  3481*  3486*  3540*  3559*  3563*  3575*  3580*  3651*  3654*  3665*
                    3685*  3702*  3723*  3728*  3745*  3765*  3775*  3779*  3791*  3796*  3869*  3872*  3883*
                    3903*  3920*  3941*  3946*  3963*  3983*  3993*  3997*  4009*  4014*  4087*  4090*  4095*
                    4102*  4122*  4147*  4155*  4156*  4181*  4191*  4195*  4207*  4212*  4236*  4304*  4307*
                    4312*  4319*  4339*  4364*  4372*  4373*  4398*  4408*  4412*  4424*  4429*  4453*  4519*
                    4522*  4527*  4534*  4554*  4583*  4587*  4599*  4604*  4628*  4633*  4650*  4712*  4716*
                    4725*  4728*  4739*  4755*  4765*  4769*  4781*  4786*  4849*  4853*  4862*  4865*  4876*
                    4892*  4902*  4906*  4918*  4923*  4993*  4997*  5006*  5010*  5024*  5034*  5037*  5049*
```

```
                    5066*  5088*  5098*  5102*  5114*  5119*  5188*  5192*  5201*  5205*  5219*  5229*  5232*
                    5244*  5261*  5283*  5293*  5297*  5309*  5314*  5381*  5385*  5394*  5397*  5408*  5419*
                    5447*  5457*  5461*  5473*  5478*  5544*  5548*  5557*  5560*  5571*  5582*  5610*  5620*
                    5624*  5636*  5641*  5711*  5715*  5724*  5727*  5738*  5753*  5782*  5792*  5795*  5810*
                    5820*  5824*  5836*  5841*  5910*  5914*  5923*  5926*  5937*  5952*  5981*  5991*  5994*
                    6009*  6019*  6023*  6035*  6040*  6106*  6110*  6118*  6125*  6133*  6140*  6172*  6182*
                    6185*  6200*  6210*  6214*  6226*  6231*  6255*  6277*  6287*  6290*  6305*  6315*  6319*
                    6331*  6336*  6399*  6403*  6407*  6426*  6430*  6442*  6447*  6511*  6515*  6523*  6538*
                    6550*  6614*  6618*  6630*  6635*  6659*  6679*  6729*  6733*  6741*  6756*  6768*  6832*
                    6836*  6848*  6853*  6877*  6897*  6954*  6958*  6966*  6994*  6995*  7004*  7016*  7038*
                    7048*  7052*  7064*  7069*  7139*  7143*  7151*  7179*  7180*  7189*  7201*  7223*  7233*
                    7237*  7249*  7254*  7321*  7325*  7334*  7361*  7378*  7382*  7394*  7399*  7473*  7477*
                    7486*  7513*  7530*  7534*  7546*  7551*  7631*  7635*  7644*  7648*  7655*  7709*  7719*
                    7723*  7735*  7740*  7812*  7816*  7825*  7829*  7836*  7890*  7900*  7904*  7916*  7921*
                    7990*  7994*  7998*  8031*  8057*  8067*  8071*  8083*  8088*  8153*  8157*  8161*  8194*
                    8220*  8230*  8234*  8246*  8251*  8326*  8330*  8337*  8344*  8347*  8353*  8358*  8364*
                    8374*  8377*  8392*  8402*  8406*  8418*  8423*  8545*  8549*  8556*  8563*  8566*  8572*
                    8577*  8583*  8593*  8596*  8611*  8621*  8625*  8637*  8642*  8753*  8757*  8761*  8764*
                    8773*  8794*  8804*  8808*  8820*  8825*  8879*  8883*  8887*  8890*  8899*  8920*  8930*
                    8934*  8946*  8951*
RHDA  = 000006      1413#  2293   2387   2543   2637
RHDB  = 000022      1419#
RHDS1 = 000012      1415#  2039   2060   2114   2115   2129   2130   2146   2147   2161   2162   2254*  2258
                    2260   2268   2300   2328   2332   2336   2504*  2508   2510   2518   2550   2578   2582
                    2586   2760   2761   2772   2773   2812   2813   2832   2847   2851   2855   2955   2956
                    2967   2968   3007   3008   3027   3042   3046   3050   3139*  3153   3157   3161   3201
                    3202   3218   3219   3241   3300*  3314   3318   3322   3362   3363   3379   3380   3402
                    3462   3466   3470   3556   3560   3564   3653*  3657   3659   3667   3688   3689   3705
                    3706   3731   3732   3748   3749   3772   3776   3780   3871*  3875   3877   3885   3906
                    3907   3923   3924   3949   3950   3966   3967   3990   3994   3998   4091   4096   4105
                    4106   4125   4126   4149*  4162   4163   4188   4192   4196   4239   4240   4308   4313
                    4322   4323   4342   4343   4366*  4379   4380   4405   4409   4413   4456   4457   4523
                    4528   4537   4538   4557   4558   4580   4584   4588   4636   4637   4653   4654   4713*
                    4717*  4731   4733   4741   4762   4766   4770   4850*  4854*  4868   4870   4878   4899
                    4903   4907   4994*  4998*  5008*  5016*  5032   5036   5039   5052   5053   5069   5070
                    5095   5099   5103   5189*  5193*  5203*  5211*  5227   5231   5234   5247   5248   5264
                    5265   5290   5294   5298   5382*  5386*  5400   5402   5410   5454   5458   5462   5545*
                    5549*  5563   5565   5573   5617   5621   5625   5712*  5716*  5726*  5730   5732   5740
                    5761   5762   5790   5794   5797   5817   5821   5825   5911*  5915*  5925*  5929   5931
                    5939   5960   5961   5989   5993   5996   6016   6020   6024   6107*  6111*  6131   6135
                    6142   6153   6154   6180   6184   6187   6207   6211   6215   6258   6259   6285   6289
                    6292   6312   6316   6320   6400*  6404*  6423   6427   6431   6512*  6516*  6525*  6544
                    6557   6558   6590   6591   6611   6615   6619   6662   6663   6680*  6730*  6734*  6743*
                    6762   6775   6776   6808   6809   6829   6833   6837   6880   6881   6898*  6955*  6959*
                    6968*  7007   7008   7045   7049   7053   7140*  7144*  7153*  7192   7193   7230   7234
                    7238   7322*  7326*  7336*  7360   7363   7375   7379   7383   7474*  7478*  7488*  7512
                    7515   7527   7531   7535   7632*  7636*  7646*  7654*  7672   7673   7692   7693   7716
                    7720   7724   7813*  7817*  7827*  7835*  7853   7854   7873   7874   7897   7901   7905
                    799'%  7995*  8010   8017   8018   8038   8039   8064   8068   8072   8154*  8158*  8173
                    8180   8181   8201   8202   8227   8231   8235   8327*  8331*  8339*  8372   8376   8379
                    8399   8403   8407   8546*  8550*  8558*  8591   8595   8598   8618   8622   8626   8754*
                    8758*  8765   8776*  8801   8805   8809   8880*  8884*  8891   8902*  8927   8931   8935
RHDT  = 000026      1421#  2086   2087   2098   2099   2292   2392   2542   2642
RHEC1 = 000044      1428#  2284   2432   2534   2682
RHEC2 = 000046      1429#  2283   2437   2533   2687
RHER1 = 000014      1416#  2295   2377   2545   2627   4088*  4089*  4093*  4094*  4305*  4306*  4310*  4311*
```

```
                    4520*  4521*  4525*  4526*  6531*  6576   6577   6749*  6794   6795   6974*  7019   7020
                    7159*  7204   7205   8341*  8351*  8355*  8457   8458   8560*  8570*  8574*  8676   8677
                    8762*  8763*  8888*  8889*
RHER2 = 000040       1426#  2290   2402   2540   2652   8342*  8350*  8356*  8471   8472   8561*  8569*  8575*
                    8690   8691
RHER3 = 000042       1427#  2285   2427   2535   2677   8343*  8349*  8357*  8485   8496   8562*  8568*  8576*
                    8704   8705
RHLA  = 000020       1418#  2291   2397   2541   2647
RHMR  = 000024       1420#  2294   2382   2544   2632
RHOF  = 000032       1423#  2289   2407   2539   2657   2806*  3001*
RHSN  = 000030       1422#  2180   2182   2286   2422   2536   2672
RHWC  = 000002       1411#
RMR   = 000004       1286#
RP6   = 000300       1136#
R0    =%000000       1118#  1941*  1984*  1985*  1986   1989   2002*  2037*  2039   2041   2043   2057*  2058*
                    2060   2062   2064   2078*  2083*  2086   2088   2095*  2098   2100   2111*  2114   2116
                    2129   2131   2143*  2146   2148   2161   2163   2179*  2180   2181*  2182   2252*  2254*
                    2255*  2258   2259   2266*  2268   2281*  2283   2284   2285   2286   2287   2288   2289
                    2290   2291   2292   2293   2294   2295   2300   2329   2331*  2332   2335*  2336   2347*
                    2352*  2376   2381   2386   2391   2396   2401   2406   2411   2416   2421   2426   2431
                    2436   2502*  2504*  2505*  2508   2509   2516*  2518   2531*  2533   2534   2535   2536
                    2537   2538   2539   2540   2541   2542   2543   2544   2545   2550   2579   2581*  2582
                    2585*  2586   2597*  2602*  2626   2631   2636   2641   2646   2651   2656   2661   2666
                    2671   2676   2681   2686   2739*  2752*  2757*  2760   2762   2769*  2772   2774   2787
                    2789   2805*  2806*  2812   2814   2826*  2832   2848   2850*  2851   2854*  2855   2866*
                    2871*  2934*  2947*  2952*  2955   2957   2964*  2967   2969   2982   2984   3000*  3001*
                    3007   3009   3021*  3027   3043   3045*  3046   3049*  3050   3061*  3066*  3137*  3139*
                    3146*  3148*  3154   3156*  3157   3160*  3161   3172*  3177*  3198*  3201   3203   3215*
                    3218   3220   3239*  3241   3298*  3300*  3307*  3309*  3315   3317*  3318   3321*  3322
                    3333*  3338*  3359*  3362   3364   3376*  3379   3381   3400*  3402   3446*  3453*  3463
                    3465*  3466   3469*  3470   3481*  3486*  3540*  3547*  3557   3559*  3560   3563*  3564
                    3575*  3580*  3651*  3653*  3654*  3657   3658   3665*  3667   3680*  3685*  3688   3690
                    3702*  3705   3707   3723*  3728*  3731   3733   3745*  3748   3750   3765*  3767*  3773
                    3775*  3776   3779*  3780   3791*  3796*  3800*  3801*  3869*  3871*  3872*  3875   3876
                    3883*  3885   3898*  3903*  3906   3908   3920*  3923   3925   3941*  3946*  3949   3951
                    3963*  3966   3968   3983*  3985*  3991   3993*  3994   3997*  3998   4009*  4014*  4018*
                    4019*  4087*  4088*  4089*  4090*  4091   4093*  4094*  4095*  4096   4102*  4105   4107
                    4122*  4125   4127   4147*  4149*  4155*  4156*  4162   4164   4181*  4183*  4189   4191*
                    4192   4195*  4196   4207*  4212*  4216*  4217*  4236*  4239   4241   4304*  4305*  4306*
                    4307*  4308   4310*  4311*  4312*  4313   4319*  4322   4324   4339*  4342   4344   4364*
                    4366*  4372*  4373*  4379   4381   4398*  4400*  4406   4408*  4409   4412*  4413   4424*
                    4429*  4433*  4434*  4453*  4456   4458   4519*  4520*  4521*  4522*  4523   4525*  4526*
                    4527*  4528   4534*  4537   4539   4554*  4557   4559   4581   4583*  4584   4587*  4588
                    4599*  4604*  4608*  4609*  4628*  4633*  4636   4638   4650*  4653   4655   4712*  4713*
                    4714*  4715*  4716*  4717*  4718*  4719*  4725*  4727   4728*  4731   4732   4739*  4741
                    4755*  4757*  4763   4765*  4766   4769*  4770   4781*  4786*  4790*  4791*  4849*  4850*
                    4851*  4852*  4853*  4854*  4855*  4856*  4862*  4864   4865*  4868   4869   4876*  4878
                    4892*  4894*  4900   4902*  4903   4906*  4907   4918*  4923*  4927*  4928*  4993*  4994*
                    4995*  4996*  4997*  4998*  4999*  5000*  5006*  5008*  5010*  5016*  5024*  5026*  5033
                    5034*  5036   5037*  5039   5049*  5052   5054   5066*  5069   5071   5088*  5090*  5096
                    5098*  5099   5102*  5103   5114*  5119*  5123*  5124*  5188*  5189*  5190*  5191*  5192*
                    5193*  5194*  5195*  5201*  5203*  5205*  5211*  5219*  5221*  5228   5229*  5231   5232*
                    5234   5244*  5247   5249   5261*  5264   5266   5283*  5285*  5291   5293*  5294   5297*
                    5298   5309*  5314*  5318*  5319*  5381*  5382*  5383*  5384*  5385*  5386*  5387*  5388*
                    5394*  5396   5397*  5400   5401   5408*  5410   5419*  5426   5428   5447*  5449*  5455
                    5457*  5458   5461*  5462   5473*  5478*  5482*  5483*  5544*  5545*  5546*  5547*  5548*
```

```
                5549*  5550*  5551*  5557*  5559   5560*  5563   5564   5571*  5573   5582*  5589   5591
                5610*  5612*  5618   5620*  5621   5624*  5625   5636*  5641*  5645*  5646*  5711*  5712*
                5713*  5714*  5715*  5716*  5717*  5718*  5724*  5726*  5727*  5730   5731   5738*  5740
                5753*  5755*  5761   5763   5769   5773*  5782*  5784*  5791   5792*  5794   5795*  5797
                5810*  5812*  5818   5820*  5821   5824*  5825   5836*  5841*  5845*  5846*  5910*  5911*
                5912*  5913*  5914*  5915*  5916*  5917*  5923*  5925*  5926*  5929   5930   5937*  5939
                5952*  5954*  5960   5962   5968   5972*  5981*  5983*  5990   5991*  5993   5994*  5996
                6009*  6011*  6017   6019*  6020   6023*  6024   6035*  6040*  6044*  6045*  6106*  6107*
                6108*  6109*  6110*  6111*  6112*  6113*  6118*  6124*  6125*  6131   6133*  6135   6140*
                6142   6153   6155   6172*  6174*  6181   6182*  6184   6185*  6187   6200*  6202*  6208
                6210*  6211   6214*  6215   6226*  6231*  6235*  6236*  6255*  6258   6260   6277*  6279*
                6286   6287*  6289   6290*  6292   6305*  6307*  6313   6315*  6316   6319*  6320   6331*
                6336*  6340*  6341*  6399*  6400*  6401*  6402*  6403*  6404*  6405*  6406*  6407*  6414*
                6424   6426*  6427   6430*  6431   6442*  6447*  6451*  6452*  6511*  6512*  6513*  6514*
                6515*  6516*  6517*  6518*  6523*  6525*  6531*  6538*  6544   6550*  6557   6559   6576
                6578   6590   6592   6612   6614*  6615   6618*  6619   6630*  6635*  6639*  6640*  6659*
                6662   6664   6679*  6680*  6681*  6682*  6729*  6730*  6731*  6732*  6733*  6734*  6735*
                6736*  6741*  6743*  6749*  6756*  6762   6768*  6775   6777   6794   6796   6808   6810
                6830   6832*  6833   6836*  6837   6848*  6853*  6857*  6858*  6877*  6880   6882   6897*
                6898*  6899*  6900*  6954*  6955*  6956*  6957*  6958*  6959*  6960*  6961*  6966*  6968*
                6974*  6975*  6977   6979   6994*  6995*  6997*  7004*  7007   7009   7016*  7019   7021
                7032*  7038*  7040*  7046   7048*  7049   7052*  7053   7064*  7069*  7073*  7074*  7139*
                7140*  7141*  7142*  7143*  7144*  7145*  7146*  7151*  7153*  7159*  7160*  7162   7164
                7179*  7180*  7182*  7189*  7192   7194   7201*  7204   7206   7217*  7223*  7225*  7231
                7233*  7234   7237*  7238   7249*  7254*  7258*  7259*  7321*  7322*  7323*  7324*  7325*
                7326*  7327*  7328*  7334*  7336*  7360   7361*  7363   7376   7378*  7379   7382*  7383
                7394*  7399*  7403*  7404*  7473*  7474*  7475*  7476*  7477*  7478*  7479*  7480*  7486*
                7488*  7512   7513*  7515   7528   7530*  7531   7534*  7535   7546*  7551*  7555*  7556*
                7631*  7632*  7633*  7634*  7635*  7636*  7637*  7638*  7644*  7646*  7648*  7654*  7655*
                7661*  7672   7674   7692   7694   7709*  7711*  7717   7719*  7720   7723*  7724   7735*
                7740*  7744*  7745*  7812*  7813*  7814*  7815*  7816*  7817*  7818*  7819*  7825*  7827*
                7829*  7835*  7836*  7842*  7853   7855   7873   7875   7890*  7892*  7898   7900*  7901
                7904*  7905   7916*  7921*  7925*  7926*  7990*  7991*  7992*  7993*  7994*  7995*  7996*
                7997*  7998*  8005*  8010   8017   8019   8031*  8038   8040   8057*  8059*  8065   8067*
                8068   8071*  8072   8083*  8088*  8092*  8093*  8153*  8154*  8155*  8156*  8157*  8158*
                8159*  8160*  8161*  8168*  8173   8180   8182   8194*  8201   8203   8220*  8222*  8228
                8230*  8231   8234*  8235   8246*  8251*  8255*  8256*  8326*  8327*  8328*  8329*  8330*
                8331*  8332*  8333*  8337*  8339*  8341*  8342*  8343*  8344*  8347*  8349*  8350*  8351*
                8352*  8353*  8355*  8356*  8357*  8358*  8364*  8366*  8373   8374*  8376   8377*  8379
                8392*  8394*  8400   8402*  8403   8406*  8407   8418*  8423*  8427*  8428*  8457   8459
                8465   8469*  8471   8473   8479   8483*  8485   8487   8493   8497*  8545*  8546*  8547*
                8548*  8549*  8550*  8551*  8552*  8556*  8558*  8560*  8561*  8562*  8563*  8566*  8568*
                8569*  8570*  8571*  8572*  8574*  8575*  8576*  8577*  8583*  8585*  8592   8593*  8595
                8596*  8598   8611*  8613*  8619   8621*  8622   8625*  8626   8637*  8642*  8646*  8647*
                8676   8678   8684   8688*  8690   8692   8698   8702*  8704   8706   8712   8716*  8753*
                8754*  8755*  8756*  8757*  8758*  8759*  8760*  8761*  8762*  8763*  8764*  8765   8768
                8773*  8776*  8777   8794*  8796*  8802   8804*  8805   8808*  8809   8820*  8825*  8829*
                8830*  8879*  8880*  8881*  8882*  8883*  8884*  8885*  8886*  8887*  8888*  8889*  8890*
                8891   8894   8899*  8902*  8903   8920*  8922*  8928   8930*  8931   8934*  8935   8946*
                8951*  8955*  8956*  9063*  9065   9067   9070   9171   9172*  9173*  9180*  9181*  9182*
                9183*  9184*  9185   9190   9196   9198*  9199   9213*  9244   9245*  9246   9249*  9357
                9367*  9371   9387   9388   9401*  9505   9509*  9510   9513   9533*  9536*  9561   9586*
                9597   9598*  9599   9600*  9601*  9602*
R1      =%000001  1119#  1928*  1929   1952*  1955   1959*  1960*  1961   8986*  8987*  8988*  8995*  8996*
                8997*  9195   9196*  9201   9205   9207   9212*  9358   9371*  9372   9376   9400*  9506
                9511*  9519*  9521*  9523*  9526*  9529   9532*  9562   9585*
```

```
R2      =%000002        1120#   9359    9370*   9374*   9377    9384*   9385*   9386    9391*   9399*   9507    9512*   9520*
                        9522*   9524*   9530    9531*   9563    9584*
R3      =%000003        1121#   9299    9308*   9314*   9315*   9318*   9323*   9324*   9325    9334*   9360    9368*   9369*
                        9383*   9386*   9395*   9396*   9398*   9435    9437*   9438    9441*   9442    9449*   9450    9452
                        9460    9464    9466*   9472    9474    9476*   9479*   9564    9583*
R4      =%000004        1122#   9300    9302*   9303*   9304*   9305    9306*   9320    9322*   9330*   9333*   9565    9582*
R5      =%000005        1123#   2308*   2558*   9019    9026*   9301    9307*   9309*   9311*   9312*   9313*   9314    9332*
                        9361    9363*   9365*   9372*   9376*   9391    9397*   9566    9581*
R6      =%000006        1124#   1126    1884*   1885*   1886    1942*
R7      =%000007        1125#   1127
SAVREG= 104420          9621#
SC      = 100000        1261#
SC1     = 000100        1336#
SC10    = 001000        1339#
SC2     = 000200        1337#
SC20    = 002000        1340#
SC4     = 000400        1338#
SEIZPT  001230          1513#   2253*   2503*   2747*   2942*   3138*   3299*   3448*   3542*   3652*   3870*   4148*   4365*
                        4726*   4863*   5007*   5202*   5395*   5558*   5725*   5924*   6119*   6167*   6272*   6524*   6742*
                        6967*   7152*   7335*   7487*   7645*   7826*   8000*   8163*   8338*   8557*   8775*   8901*  10282
                       10285   10288   10290   10292   10294   10301   10303
SKI     = 040000        1395#
SP      =%000006        1126#   1888*   1907    1915*   1916*   1917    1919*   1922*   1952    1971*   1975*   1980    1982
                        1989*   2283*   2284*   2285*   2286*   2287*   2288*   2289*   2290*   2291*   2292*   2293*   2294*
                        2295*   2310    2311    2378    2383    2388    2393    2398    2403    2408    2413    2418    2423
                        2428    2433    2438    2533*   2534*   2535*   2536*   2537*   2538*   2539*   2540*   2541*   2542*
                        2543*   2544*   2545*   2560    2561    2628    2633    2638    2643    2648    2653    2658    2663
                        2668    2673    2678    2683    2688    8992*   9000*   9001*   9017*   9018*   9019*   9020*   9021*
                        9022*   9023*   9024*   9025*   9060*   9098*   9101    9103    9104    9121    9123*   9149    9171*
                        9176*   9195*   9201*   9205*   9212    9213    9244*   9245    9246*   9248    9249    9250*   9253
                        9255*   9257*   9263    9291*   9292    9293    9294*   9299*   9300*   9301*   9307    9332    9333
                        9334    9335*   9336*   9357*   9358*   9359*   9360*   9361*   9362*   9363    9366*   9379    9381*
                        9383    9393    9395    9397    9398    9399    9400    9401    9403*   9404*   9421*   9422*   9425*
                        9426*   9435*   9436*   9441    9444    9448*   9455    9459*   9478    9479    9480*   9481*   9482*
                        9503*   9504*   9505*   9506*   9507*   9509    9513*   9515    9517    9525*   9526    9528    9529*
                        9531    9532    9533    9535    9561*   9562*   9563*   9564*   9565*   9566*   9567*   9568*   9569*
                        9570*   9577*   9578*   9579*   9580*   9581    9582    9583    9584    9585    9586    9597*   9598
STACK   = 001100        1107#   1888    1942    1971
START   = 002044        1442    1882#   1992
START1  = 002200        1902#   1903*   1904*
STKLMT= 177774          1112#
SWR     = 177570        1114#   1115    2186    2195    2447    2697    2895    3090    3252    3413    3509    3603    3822
                        4040    4259    4476    4673    4810    4947    5143    5338    5502    5665    5865    6064    6360
                        6470    6689    6907    7093    7278    7431    7583    7765    7946    8111    8274    8446    8665
                        8784    8910    9093*   9111    9145    9152    9156
SW0     = 000001        1165#
SW00    = 000001        1155#   1165
SW01    = 000002        1154#   1164
SW02    = 000004        1153#   1163
SW03    = 000010        1152#   1162
SW04    = 000020        1151#   1161
SW05    = 000040        1150#   1160
SW06    = 000100        1149#   1159
SW07    = 000200        1148#   1158
SW08    = 000400        1147#   1157
SW09    = 001000        1146#   1156    2195    2447    2697    2895    3090    3252    3413    3509    3603    3822    4040
```

```
                        4259    4476    4673    4810    4947    5143    5338    5502    5665    5865    6064    6360    6470
                        6689    6907    7093    7278    7431    7583    7765    7946    8111    8274    8446    8665    8784
                        8910
SW1     = 000002        1164#
SW10    = 002000        1145#
SW11    = 004000        1144#
SW12    = 010000        1143#
SW13    = 020000        1142#
SW14    = 040000        1141#
SW15    = 100000        1140#   2186
SW2     = 000004        1163#
SW3     = 000010        1162#
SW4     = 000020        1161#
SW5     = 000040        1160#
SW6     = 000100        1159#
SW7     = 000200        1158#
SW8     = 000400        1157#
SW9     = 001000        1156#
TBITVE= 000014          1198#
TDF     = 000040        1354#
TESTNO  062207          1950    9663#
TEST1   002764          2025    2026    2033#
TEST10  013772          3437    3438    3445#
TEST11  014442          3532    3533    3539#
TEST12  015112          3637    3638    3645#
TEST13  016514          3855    3856    3863#
TEST14  020116          4074    4075    4082#
TEST15  021446          4291    4292    4299#
TEST16  022776          4506    4507    4514#
TEST17  024264          4700    4701    4708#
TEST2   004316          2228    2229    2236#
TEST20  025176          4836    4837    4845#
TEST21  026110          4981    4982    4989#
TEST22  027314          5176    5177    5184#
TEST23  030520          5369    5370    5377#
TEST24  031554          5532    5533    5540#
TEST25  032610          5699    5700    5707#
TEST26  034024          5898    5899    5906#
TEST27  035240          6094    6095    6102#
TEST3   005742          2479    2480    2486#
TEST30  037256          6387    6388    6395#
TEST31  040000          6499    6500    6507#
TEST32  041266          6717    6718    6725#
TEST33  042554          6942    6943    6950#
TEST34  043672          7127    7128    7135#
TEST35  045010          7309    7310    7317#
TEST36  045652          7461    7462    7469#
TEST37  046514          7619    7620    7627#
TEST4   007366          2730    2731    2738#
TEST40  047510          7800    7801    7808#
TEST41  050504          7978    7979    7986#
TEST42  051514          8141    8142    8149#
TEST43  052524          8314    8315    8322#
TEST44  054112          8533    8534    8541#
TEST45  055500          8741    8742    8749#
TEST46  056416          8867    8868    8875#
```

```
TEST5   010552          2925    2926    2933#
TEST6   011736          3117    3118    3125#
TEST7   012754          3278    3279    3286#
TIME    001244          1519#   2245*   2307    2316    2495*   2557    2566    2745*   2940*   3130*   3291*   6536*   6754*
                        7346*   7354*   7368    7498*   7506*   7520    9007*   10303
TIMEA   001250          1521#   2238*   2307*   2309
TIMEAM  001254          1523#   2240*   2311*   7423
TIMEAP  001252          1522#   2239*   2310*   7421    7666
TIMEB   001256          1524#   2488*   2557*   2559
TIMEBM  001262          1526#   2490*   2561*   7575
TIMEBP  001260          1525#   2489*   2560*   7573    7847
TIMES   001264          1527#   7368*   7421    7423    7520*   7573    7575
TITLE   061714          1902    9630#
TKVEC = 000060          1205#
TOLER   057450          2308    2558    9017#
TPVEC = 000064          1206#
TRAPVE= 000034          1204#   1893*   1894*   1972*   1973*
TRE   = 040000          1260#   5773    5972    8469    8483    8497    8688    8702    8716
TRK1  = 004000          1341#
TRK10 = 040000          1344#
TRK2  = 010000          1342#
TRK20 = 100000          1345#
TRK4  = 020000          1343#
TRTVEC= 000014          1199#
TSTADR  070520          1961    10331#
TSTERR  062255          9670#
TSTNUM  001234          1515#   9141*   10273   10275   10277   10280   10282   10285   10288   10290   10292   10294   10296
                        10298   10299   10301   10303
TST1    002710          1954    2017#   10331
TST1AA  002704          2002#   9074
TST10   013716          3412    3414    3429#   10338
TST11   014366          3508    3510    3524#   10339
TST12   015036          3602    3604    3629#   10340
TST13   016440          3821    3823    3847#   10341
TST14   020042          4039    4041    4066#   10342
TST15   021372          4258    4260    4283#   10343
TST16   022722          4475    4477    4498#   10344
TST17   024210          4672    4674    4692#   10345
TST2    004242          2194    2196    2220#   10332
TST20   025122          4809    4811    4828#   10346
TST21   026034          4946    4948    4973#   10347
TST22   027240          5142    5144    5168#   10348
TST23   030444          5337    5339    5361#   10349
TST24   031500          5501    5503    5524#   10350
TST25   032534          5664    5666    5691#   10351   10352
TST26   033750          5864    5866    5890#   10353
TST27   035164          6063    6065    6086#   10354
TST3    005666          2446    2448    2471#   10333
TST30   037202          6359    6361    6379#   10355
TST31   037724          6469    6471    6491#   10356
TST32   041212          6688    6690    6709#   10357
TST33   042500          6906    6908    6934#   10358
TST34   043616          7092    7094    7119#   10359
TST35   044734          7277    7279    7301#   10360
TST36   045576          7430    7432    7453#   10361
TST37   046440          7582    7584    7611#   10362
```

```
TST4     007312          2696    2698    2722#   10334
TST40    047434          7764    7766    7792#   10363
TST41    050430          7945    7947    7970#   10364
TST42    051440          8110    8112    8133#   10365
TST43    052450          8273    8275    8306#   10366
TST43B   053550          8346    8360    8455#
TST44    054036          8445    8447    8451    8525#   10367
TST44B   055136          8565    8579    8674#
TST45    055424          8664    8666    8670    8733#   10368
TST46    056342          8859#   10369
TST5     010476          2894    2896    2917#   10335
TST6     011662          3089    3091    3109#   10336
TST7     012700          3251    3253    3270#   10337
TUF   =  000100          1355#
TYPDS =  104410          1920    1923    9061    9206    9617#
TYPE  =  104400          1902    1905    1910    1918    1921    1924    1932    1940    1950    1957    1974    1977    1978
                         1988    9059    9062    9147    9155    9170    9187    9189    9192    9194    9209    9214    9326
                         9402    9447    9453    9458    9462    9467    9468    9470    9473    9477    9537    9539    9613#
TYPOC =  104402          1976    1990    9178    9202    9614#
TYPON =  104406          9616#
TYPOS =  104404          9615#
UNS   =  040000          1298#
UPE   =  020000          1236#
US1   =  000001          1223#
US2   =  000002          1224#
US4   =  000004          1225#
UWR   =  000010          1391#
VUF   =  000002          1390#
VU30  =  010000          1361#
VV    =  000100          1271#   2334    2338    2359    2365    2584    2588    2609    2615    2815    2853    2857    2878
                         2884    3010    3044    3048    3052    3155    3159    3163    3316    3320    3324    3464    3468
                         3472    3558    3562    3566    3668    3708    3751    3774    3778    3782    3886    3926    3969
                         3992    3996    4000    4190    4194    4198    4407    4411    4415    4586    4590    4742    4764
                         4768    4772    4879    4901    4905    4909    5031    5097    5101    5105    5226    5292    5296
                         5300    5411    5456    5460    5464    5574    5619    5623    5627    5741    5789    5819    5823
                         5827    5940    5988    6018    6022    6026    6156    6179    6209    6213    6217    6261    6284
                         6314    6318    6322    6425    6429    6433    6617    6621    6835    6839    7047    7051    7055
                         7232    7236    7240    7377    7381    7385    7529    7533    7537    7718    7722    7726    7899
                         7903    7907    8066    8070    8074    8229    8233    8237    8371    8401    8405    8409    8590
                         8620    8624    8628    8803    8807    8811    8929    8933    8937
VVSET =  000001          1876#   2269    2330    2356    2359    2365    2519    2580    2606    2609    2615    2849    2875
                         2878    2884    3002#   3044    3070    3073    3074    3079    3080    3155    3181    3184    3185
                         3190    3191    3316    3342    3345    3346    3351    3352    3464    3490    3493    3494    3499
                         3500    3558    3584    3587    3588    3593    3594    3668    3774    3800    3802    3805    3806
                         3811    3812    3886    3992    4018    4020    4023    4024    4029    4030    4190    4216    4218
                         4221    4222    4227    4228    4407    4433    4435    4438    4439    4444    4445    4608    4610
                         4613    4618    4742    4764    4790    4792    4795    4800    4879    4901    4927    4929    4932
                         4937    5031    5097    5123    5125    5128    5133    5226    5292    5318    5320    5323    5328
                         5411    5456    5482    5484    5487    5492    5574    5619    5645    5647    5650    5655    5741
                         5789    5819    5845    5847    5850    5855    5940    5988    6018    6044    6046    6049    6054
                         6179    6209    6235    6237    6240    6245    6284    6314    6340    6342    6345    6350    6425
                         6451    6456    6461    6639    6641    6644    6645    6650    6651    6857    6859    6862    6863
                         6868    6869    7047    7073    7075    7078    7083    7232    7258    7260    7263    7268    7377
                         7403    7408    7413    7529    7555    7560    7565    7718    7744    7746    7749    7754    7899
                         7925    7927    7930    7935    8066    8092    8094    8097    8102    8229    8255    8257    8260
                         8265    8371    8401    8427    8429    8432    8437    8590    8620    8646    8648    8651    8656
```

```
                    8803    8829    8831    8834    8835    8840    8841    8929    8955    8957    8960    8961    8966
                    8967
WATCH   001246      1520#   2246*   2302    2496*   2552    2746*   2834    2941*   3029    3131*   3243    3292*   3404
                    6537*   6546    6755*   6764    7347*   7348    7355*   7365    7499*   7500    7507*   7517    7666*
                    7684    7847*   7865    9008    9010*   9012*
WCE   = 040000      1237#
WCF   = 000040      1289#
WCU   = 000001      1349#
WLE   = 004000      1295#
WRL   = 004000      1276#
WRU   = 000400      1357#
WSC   = 000004      1351#
$BDADR  001122      1477#   2042*   2043*   2063*   2064*   2087*   2088*   2099*   2100*   2115*   2116*   2130*   2131*
                    2147*   2148*   2162*   2163*   2259*   2260*   2328*   2329*   2376*   2377*   2381*   2382*   2386*
                    2387*   2391*   2392*   2396*   2397*   2401*   2402*   2406*   2407*   2411*   2412*   2416*   2417*
                    2421*   2422*   2426*   2427*   2431*   2432*   2436*   2437*   2509*   2510*   2578*   2579*   2626*
                    2627*   2631*   2632*   2636*   2637*   2641*   2642*   2646*   2647*   2651*   2652*   2656*   2657*
                    2661*   2662*   2666*   2667*   2671*   2672*   2676*   2677*   2681*   2682*   2686*   2687*   2761*
                    2762*   2773*   2774*   2788*   2789*   2813*   2814*   2847*   2848*   2956*   2957*   2968*   2969*
                    2983*   2984*   3008*   3009*   3042*   3043*   3153*   3154*   3202*   3203*   3219*   3220*   3314*
                    3315*   3363*   3364*   3380*   3381*   3462*   3463*   3556*   3557*   3658*   3659*   3689*   3690*
                    3706*   3707*   3732*   3733*   3749*   3750*   3772*   3773*   3876*   3877*   3907*   3908*   3924*
                    3925*   3950*   3951*   3967*   3968*   3990*   3991*   4106*   4107*   4126*   4127*   4163*   4164*
                    4188*   4189*   4240*   4241*   4323*   4324*   4343*   4344*   4380*   4381*   4405*   4406*   4457*
                    4458*   4538*   4539*   4558*   4559*   4580*   4581*   4637*   4638*   4654*   4655*   4732*   4733*
                    4762*   4763*   4869*   4870*   4899*   4900*   5032*   5033*   5053*   5054*   5070*   5071*   5095*
                    5096*   5227*   5228*   5248*   5249*   5265*   5266*   5290*   5291*   5401*   5402*   5427*   5428*
                    5454*   5455*   5564*   5565*   5590*   5591*   5617*   5618*   5731*   5732*   5762*   5763*   5790*
                    5791*   5817*   5818*   5930*   5931*   5961*   5962*   5989*   5990*   6016*   6017*   6154*   6155*
                    6180*   6181*   6207*   6208*   6259*   6260*   6285*   6286*   6312*   6313*   6423*   6424*   6558*
                    6559*   6577*   6578*   6591*   6592*   6611*   6612*   6663*   6664*   6776*   6777*   6795*   6796*
                    6809*   6810*   6829*   6830*   6881*   6882*   6978*   6979*   7008*   7009*   7020*   7021*   7045*
                    7046*   7163*   7164*   7193*   7194*   7205*   7206*   7230*   7231*   7375*   7376*   7527*   7528*
                    7673*   7674*   7693*   7694*   7716*   7717*   7854*   7855*   7874*   7875*   7897*   7898*   8018*
                    8019*   8039*   8040*   8064*   8065*   8181*   8182*   8202*   8203*   8227*   8228*   8372*   8373*
                    8399*   8400*   8458*   8459*   8472*   8473*   8486*   8487*   8591*   8592*   8618*   8619*   8677*
                    8678*   8691*   8692*   8705*   8706*   8767*   8768*   8801*   8802*   8893*   8894*   8927*   8928*
                   10273   10275   10277   10280   10282   10285   10290
$BDDAT  001126      1479#   2041*   2045    2049    2062*   2066    2070    2086*   2090    2098*   2102    2114*   2118
                    2122    2129*   2133    2137    2146*   2150    2154    2161*   2165    2169    2182*   2183    2258*
                    2262    2268*   2272    2345*   2351*   2357*   2363*   2378*   2383*   2388*   2393*   2398*   2403*
                    2408*   2413*   2418*   2423*   2428*   2433*   2438*   2508*   2512    2518*   2522    2595*   2601*
                    2607*   2613*   2628*   2633*   2638*   2643*   2648*   2653*   2658*   2663*   2668*   2673*   2678*
                    2683*   2688*   2760*   2764    2772*   2776    2780    2787*   2791    2795    2812*   2816    2820
                    2864*   2870*   2876*   2882*   2955*   2959    2967*   2971    2975    2982*   2986    2990    3007*
                    3011    3015    3059*   3065*   3071*   3077*   3170*   3176*   3182*   3188*   3201*   3205    3209
                    3218*   3222    3226    3331*   3337*   3343*   3349*   3362*   3366    3370    3379*   3383    3387
                    3479*   3485*   3491*   3497*   3573*   3579*   3585*   3591*   3657*   3661    3667*   3671    3688*
                    3692    3696    3705*   3709    3713    3731*   3735    3739    3748*   3752    3756    3789*   3795*
                    3803*   3809*   3875*   3879    3885*   3889    3906*   3910    3914    3923*   3927    3931    3949*
                    3953    3957    3966*   3970    3974    4007*   4013*   4021*   4027*   4105*   4109    4113    4125*
                    4129    4133    4162*   4166    4170    4205*   4211*   4219*   4225*   4239*   4243    4247    4322*
                    4326    4330    4342*   4346    4350    4379*   4383    4387    4422*   4428*   4436*   4442*   4456*
                    4460    4464    4537*   4541    4545    4557*   4561    4565    4597*   4603*   4611*   4616*   4636*
                    4640    4644    4653*   4657    4661    4731*   4735    4741*   4745    4779*   4785*   4793*   4798*
                    4868*   4872    4878*   4882    4916*   4922*   4930*   4935*   5039*   5044*   5046    5052*   5056
```

```
                    5060    5063*   5073    5077    5112*   5118*   5126*   5131*   5234*   5239*   5241    5247*   5251
                    5255    5264*   5268    5272    5307*   5313*   5321*   5326*   5400*   5404    5410*   5414    5426*
                    5430    5434    5471*   5477*   5485*   5490*   5563*   5567    5573*   5577    5589*   5593    5597
                    5634*   5640*   5648*   5653*   5730*   5734    5740*   5744    5761*   5765    5769*   5770    5797*
                    5802*   5804    5834*   5840*   5848*   5853*   5929*   5933    5939*   5943    5960*   5964    5968*
                    5969    5996*   6001*   6003    6033*   6039*   6047*   6052*   6153*   6157    6161    6187*   6192*
                    6194    6224*   6230*   6238*   6243*   6258*   6262    6266    6292*   6297*   6299    6329*   6335*
                    6343*   6348*   6440*   6446*   6454*   6459*   6557*   6561    6565    6576*   6580    6590*   6594
                    6598    6628*   6634*   6642*   6648*   6662*   6666    6670    6775*   6779    6783    6794*   6798
                    6808*   6812    6816    6846*   6852*   6860*   6866*   6880*   6884    6888    6977*   6981    6985
                    7007*   7011    7019*   7023    7062*   7068*   7076*   7081*   7162*   7166    7170    7192*   7196
                    7204*   7208    7247*   7253*   7261*   7266*   7392*   7398*   7406*   7411*   7544*   7550*   7558*
                    7563*   7672*   7676    7692*   7696    7733*   7739*   7747*   7752*   7853*   7857    7873*   7877
                    7914*   7920*   7928*   7933*   8017*   8021    8025    8038*   8042    8046    8081*   8087*   8095*
                    8100*   8180*   8184    8188    8201*   8205    8209    8244*   8250*   8258*   8263*   8379*   8384*
                    8386    8416*   8422*   8430*   8435*   8457*   8461    8465*   8466    8471*   8475    8479*   8480
                    8485*   8489    8493*   8494    8598*   8603*   8605    8635*   8641*   8649*   8654*   8676*   8680
                    8684*   8685    8690*   8694    8698*   8699    8704*   8708    8712*   8713    8777*   8778    8818*
                    8824*   8832*   8838*   8903*   8904    8944*   8950*   8958*   8964*   10273   10275   10277   10280
                   10282   10285   10290
$BELL   001174      1499#   9147    9161
$CMTAG  001100      1465#   1883    1884    1891    1895    1896    1897
$CM1    = 000001    1491#   1492#
$CM2    = 000002    1491#   1492#
$CM3    = 000001    1489#   1491
$CM4    = 000005    1492#   1493#   1494#   1495#   1496#   1497#
$CNTLU  061400      9462    9486#
$CRLF   001201      1501#   1924    1940    1977    9155    9161    9170    9189    9194    9214    9467    9488    9542
$DBLK   061100      9368    9402    9410#
$DOAGN  057636      9055    9064    9068    9074#
$DTBL   061070      9371    9406#
$ENDAD  057626      1459    1461    9065    9066    9070#
$ENDCT  057556      1895    9057#
$ENDMG  057642      9059    9075#
$ENULL  057657      9062    9078#
$EOP    057510      8971    9044#
$EOPCT  057550      1895*   9054#   9058
$ERFLG  001103      1468#   2193    2197*   2445    2449*   2695    2699*   2893    2897*   3088    3092*   3250    3254*
                    3411    3415*   3507    3511*   3601    3605*   3820    3824*   4038    4042*   4257    4261*   4474
                    4478*   4671    4675*   4808    4812*   4945    4949*   5141    5145*   5336    5340*   5500    5504*
                    5663    5667*   5863    5867*   6062    6066*   6358    6362*   6468    6472*   6687    6691*   6905
                    6909*   7091    7095*   7276    7280*   7429    7433*   7581    7585*   7763    7767*   7944    7948*
                    8109    8113*   8272    8276*   8444    8448*   8663    8667*   9085    9107    9109*   9126    9142*
                    9161
$ERMAX  001115      1474#   1898*   1947*   9126
$ERROR  060040      1991    9140#
$ERRPC  001116      1475#   9149*   9150*   9151    9161    9176    10273   10275   10277   10280   10282   10285   10288
                   10290   10292   10294   10296   10298   10299   10301   10303
$ERRTB  001274      1555#   9184
$ERRTY  060154      9154    9169#
$ERTTL  001112      1472#   9148*   9161
$ESCAP  001172      1498#   1897*
$FILLC  001150      1487#   9253    9265
$FILLS  001147      1486#   9265
$GDADR  001120      1476#
$GDBAT  001124      1478#   2044*   2047    2051*   2065*   2068    2072*   2089*   2090    2101*   2102    2117*   2120
```

```
                    4484#   4486    4680#   4682    4816#   4818    4954#   4956    5149#   5151    5345#   5347    5508#
                    5510    5672#   5674    5871#   5873    6071#   6073    6367#   6369    6477#   6479    6695#   6697
                    6914#   6916    7099#   7101    7285#   7287    7437#   7439    7590#   7592    7771#   7773    7953#
                    7955    8117#   8119    8281#   8283    8500#   8502    8720#   8722    8846#   8848
$OCNT   060660      9298*   9327*   9340#
$OMODE  060662      9293*   9297*   9302    9305*   9316*   9342#
$OVER   060022      9094    9105    9117    9122#
$PASS   001100      1466#   1946*   9051*   9052*   9060    9075    9113    9126
$QUES   001200      1500#   9161    9470    9488    9539    9542
$RDCHR  061110      9421#   9618
$RDDEC= ****** U    9621
$RDLIN  061144      9435#   9619
$RDOCT  061414      9503#   9620
$RDSZ = 000007      9428#
$REGAD  001152      1489#
$REGD   001154      1491#
$RESRE  061612      9576#   9622
$RPADR  001270      1536#   1941    1975    1982*   1984    2002
$RPVEC  001272      1537#
$SAVRE  061554      9560#   9621
$SCOPE  057662      1889    9092#
$SETUP= 000027      1876#   1889    1891    1893    1895    1896    1897    1899    9049
$STUP = 177777      1876#
$SVLAD  060012      9102    9120#
$SWR  = 166000      1078#   1089    1095    1096    1097    1098    1099    1100    1101    1497    1498    1499    1896
                    1897    1899    1900    2027    2230    2481    2732    2927    3119    3280    3439    3534    3639
                    3857    4076    4293    4508    4702    4838    4983    5178    5371    5534    5701    5900    6096
                    6389    6501    6719    6944    7129    7311    7463    7621    7802    7980    8143    8316    8535
                    8743    8869    9041    9050    9063    9075    9079#   9086    9087    9088    9089    9093    9105
                    9107    9108    9109    9110    9111    9122    9125    9133    9134    9135    9136    9137    9145
                    9152    9156    9159    9161
$SWRMK= 000000      9089
$TIMES  001170      1497#   1896*   2027*   2198*   2230*   2450*   2481*   2700*   2732*   2898*   2927*   3093*   3119*
                    3255*   3280*   3416*   3439*   3512*   3534*   3606*   3639*   3825*   3857*   4043*   4076*   4262*
                    4293*   4479*   4508*   4676*   4702*   4813*   4838*   4950*   4983*   5146*   5178*   5341*   5371*
                    5505*   5534*   5668*   5701*   5868*   5900*   6067*   6096*   6363*   6389*   6473*   6501*   6692*
                    6719*   6910*   6944*   7096*   7129*   7281*   7311*   7434*   7463*   7586*   7621*   7768*   7802*
                    7949*   7980*   8114*   8143*   8277*   8316*   8449*   8535*   8668*   8743*   8783*   8788    8869*
                    8909*   8914    9050*   9110*   9116    9119*   9125
$TKB    001140      1482#   9415    9425
$TKS    001136      1481#   1886    9415    9423
$TMPO   001156      1492#   2045*   2046*   2047    2066*   2067*   2068    2118*   2119*   2120    2133*   2134*   2135
                    2150*   2151*   2152    2165*   2166*   2167    2272*   2273*   2274    2333*   2334*   2339    2341
                    2348    2522*   2523*   2524    2583*   2584*   2589    2591    2598    2776*   2777*   2778    2791*
                    2792*   2793    2816*   2817*   2818    2852*   2853*   2858    2860    2867    2971*   2972*   2973
                    2986*   2987*   2988    3011*   3012*   3013    3047*   3048*   3053    3055    3062    3158*   3159*
                    3164    3166    3173    3205*   3206*   3207    3222*   3223*   3224    3319*   3320*   3325    3327
                    3334    3366*   3367*   3368    3383*   3384*   3385    3467*   3468*   3473    3475    3482    3561*
                    3562*   3567    3569    3576    3671*   3672*   3673    3692*   3693*   3694    3709*   3710*   3711
                    3735*   3736*   3737    3752*   3753*   3754    3777*   3778*   3783    3785    3792    3889*   3890*
                    3891    3910*   3911*   3912    3927*   3928*   3929    3953*   3954*   3955    3970*   3971*   3972
                    3995*   3996*   4001    4003    4010    4109*   4110*   4111    4129*   4130*   4131    4166*   4167*
                    4168    4193*   4194*   4199    4201    4208    4243*   4244*   4245    4326*   4327*   4328    4346*
                    4347*   4348    4383*   4384*   4385    4410*   4411*   4416    4418    4425    4460*   4461*   4462
                    4541*   4542*   4543    4561*   4562*   4563    4585*   4586*   4591    4593    4600    4640*   4641*
                    4642    4657*   4658*   4659    4745*   4746*   4747    4767*   4768*   4773    4775    4782    4882*
```

```
                    4883*  4884   4904*  4905*  4910   4912   4919   5036*  5041   5044   5056*  5057*  5058
                    5073*  5074*  5075   5100*  5101*  5106   5108   5115   5231*  5236   5239   5251*  5252*
                    5253   5268*  5269*  5270   5295*  5296*  5301   5303   5310   5414*  5415*  5416   5430*
                    5431*  5432   5459*  5460*  5465   5467   5474   5577*  5578*  5579   5593*  5594*  5595
                    5622*  5623*  5628   5630   5637   5744*  5745*  5746   5794*  5799   5802   5822*  5823*
                    5828   5830   5837   5943*  5944*  5945   5993*  5998   6001   6021*  6022*  6027   6029
                    6036   6157*  6158*  6159   6184*  6189   6192   6212*  6213*  6218   6220   6227   6262*
                    6263*  6264   6289*  6294   6297   6317*  6318*  6323   6325   6332   6412*  6413*  6414
                    6428*  6429*  6434   6436   6443   6561*  6562*  6563   6594*  6595*  6596   6616*  6617*
                    6622   6624   6631   6666*  6667*  6668   6779*  6780*  6781   6812*  6813*  6814   6834*
                    6835*  6840   6842   6849   6884*  6885*  6886   6981*  6982*  6983   7050*  7051*  7056
                    7058   7065   7166*  7167*  7168   7235*  7236*  7241   7243   7250   7380*  7381*  7386
                    7388   7395   7532*  7533*  7538   7540   7547   7721*  7722*  7727   7729   7736   7902*
                    7903*  7908   7910   7917   8021*  8022*  8023   8042*  8043*  8044   8069*  8070*  8075
                    8077   8084   8184*  8185*  8186   8205*  8206*  8207   8232*  8233*  8238   8240   8247
                    8376*  8381   8384   8404*  8405*  8410   8412   8419   8595*  8600   8603   8623*  8624*
                    8629   8631   8638   8778*  8779*  8780   8806*  8807*  8812   8814   8821   8904*  8905*
                    8906   8932*  8933*  8938   8940   8947
STMP1   001160      1493*  2270*  2271*  2273   2337*  2338*  2339   2353   2520*  2521*  2523   2587*  2588*
                    2589   2603   2856*  2857*  2858   2872   3051*  3052*  3053   3067   3162*  3163*  3164
                    3178   3323*  3324*  3325   3339   3471*  3472*  3473   3487   3565*  3566*  3567   3581
                    3669*  3670*  3672   3781*  3782*  3783   3797   3887*  3888*  3890   3999*  4000*  4001
                    4015   4197*  4198*  4199   4213   4414*  4415*  4416   4430   4589*  4590*  4591   4605
                    4743*  4744*  4746   4771*  4772*  4773   4787   4880*  4881*  4883   4908*  4909*  4910
                    4924   5104*  5105*  5106   5120   5299*  5300*  5301   5315   5412*  5413*  5415   5463*
                    5464*  5465   5479   5575*  5576*  5578   5626*  5627*  5628   5642   5742*  5743*  5745
                    5826*  5827*  5828   5842   5941*  5942*  5944   6025*  6026*  6027   6041   6216*  6217*
                    6218   6232   6321*  6322*  6323   6337   6432*  6433*  6434   6448   6620*  6621*  6622
                    6636   6838*  6839*  6840   6854   7054*  7055*  7056   7070   7239*  7240*  7241   7255
                    7384*  7385*  7386   7400   7536*  7537*  7538   7552   7725*  7726*  7727   7741   7906*
                    7907*  7908   7922   8073*  8074*  8075   8089   8236*  8237*  8238   8252   8408*  8409*
                    8410   8424   8627*  8628*  8629   8643   8770*  8771*  8779   8810*  8811*  8812   8826
                    8896*  8897*  8905   8936*  8937*  8938   8952
STMP2   001162      1494*  2332*  2333   2345   2357   2359*  2360   2582*  2583   2595   2607   2609*  2610
                    2851*  2852   2864   2876   2878*  2879   3046*  3047   3059   3071   3073*  3074   3157*
                    3158   3170   3182   3184*  3185   3318*  3319   3331   3343   3345*  3346   3466*  3467
                    3479   3491   3493*  3494   3560*  3561   3573   3585   3587*  3588   3776*  3777   3789
                    3803   3805*  3806   3994*  3995   4007   4021   4023*  4024   4192*  4193   4205   4219
                    4221*  4222   4409*  4410   4422   4436   4438*  4439   4584*  4585   4597   4611   4613
                    4766*  4767   4779   4793   4795   4903*  4904   4916   4930   4932   5099*  5100   5112
                    5126   5128   5294*  5295   5307   5321   5323   5458*  5459   5471   5485   5487   5621*
                    5622   5634   5648   5650   5821*  5822   5834   5848   5850   6020*  6021   6033   6047
                    6049   6211*  6212   6224   6238   6240   6316*  6317   6329   6343   6345   6427*  6428
                    6440   6454   6456   6615*  6616   6628   6642   6644*  6645   6833*  6834   6846   6860
                    6862*  6863   7049*  7050   7062   7076   7078   7234*  7235   7247   7261   7263   7379*
                    7380   7392   7406   7408   7531*  7532   7544   7558   7560   7720*  7721   7733   7747
                    7749   7901*  7902   7914   7928   7930   8068*  8069   8081   8095   8097   8231*  8232
                    8244   8258   8260   8403*  8404   8416   8430   8432   8622*  8623   8635   8649   8651
                    8805*  8806   8818   8832   8834*  8835   8931*  8932   8944   8958   8960*  8961   10299
STMP3   001164      1495*  2336*  2337   2351   2363   2365*  2366   2586*  2587   2601   2613   2615*  2616
                    2855*  2856   2870   2882   2884*  2885   3050*  3051   3065   3077   3079*  3080   3161*
                    3162   3176   3188   3190*  3191   3322*  3323   3337   3349   3351*  3352   3470*  3471
                    3485   3497   3499*  3500   3564*  3565   3579   3591   3593*  3594   3780*  3781   3795
                    3809   3811*  3812   3998*  3999   4013   4027   4029*  4030   4196*  4197   4211   4225
                    4227*  4228   4413*  4414   4428   4442   4444*  4445   4588*  4589   4603   4616   4618
                    4770*  4771   4785   4798   4800   4907*  4908   4922   4935   4937   5103*  5104   5118
```

```
                    5131  5133  5298* 5299  5313  5326  5328  5462* 5463  5477  5490  5492  5625*
                    5626  5640  5653  5655  5825* 5826  5840  5853  5855  6024* 6025  6039  6052
                    6054  6215* 6216  6230  6243  6245  6320* 6321  6335  6348  6350  6431* 6432
                    6446  6459  6461  6619* 6620  6634  6648  6650* 6651  6837* 6838  6852  6866
                    6868* 6869  7053* 7054  7068  7081  7083  7238* 7239  7253  7266  7268  7383*
                    7384  7398  7411  7413  7535* 7536  7550  7563  7565  7724* 7725  7739  7752
                    7754  7905* 7906  7920  7933  7935  8072* 8073  8087  8100  8102  8235* 8236
                    8250  8263  8265  8407* 8408  8422  8435  8437  8626* 8627  8641  8654  8656
                    8809* 8810  8824  8838  8840* 8841  8935* 8936  8950  8964  8966* 8967  10299
$TMP4   001166      1496# 2049* 2050* 2051  2070* 2071* 2072  2122* 2123* 2124  2137* 2138* 2139
                    2154* 2155* 2156  2169* 2170* 2171  2780* 2781* 2782  2795* 2796* 2797  2820*
                    2821* 2822  2975* 2976* 2977  2990* 2991* 2992  3015* 3016* 3017  3209* 3210*
                    3211  3226* 3227* 3228  3370* 3371* 3372  3387* 3388* 3389  3696* 3697* 3698
                    3713* 3714* 3715  3739* 3740* 3741  3756* 3757* 3758  3914* 3915* 3916  3931*
                    3932* 3933  3957* 3958* 3959  3974* 3975* 3976  4113* 4114* 4115  4133* 4134*
                    4135  4170* 4171* 4172  4247* 4248* 4249  4330* 4331* 4332  4350* 4351* 4352
                    4387* 4388* 4389  4464* 4465* 4466  4545* 4546* 4547  4565* 4566* 4567  4644*
                    4645* 4646  4661* 4662* 4663  5060* 5061* 5062  5077* 5078* 5079  5255* 5256*
                    5257  5272* 5273* 5274  5434* 5435* 5436  5597* 5598* 5599  6161* 6162* 6163
                    6266* 6267* 6268  6565* 6566* 6567  6598* 6599* 6600  6670* 6671* 6672  6783*
                    6784* 6785  6816* 6817* 6818  6888* 6889* 6890  6985* 6986* 6987  7170* 7171*
                    7172  8025* 8026* 8027  8046* 8047* 8048  8188* 8189* 8190  8209* 8210* 8211
$TN     = 000047    1078# 1089  2004  2017  2027# 2029  2194  2196  2202  2220  2230# 2232  2446
                    2448  2453  2471  2481# 2482  2696  2698  2704  2722  2732# 2734  2894  2896
                    2901  2917  2927# 2929  3089  3091  3097  3109  3119# 3121  3251  3253  3258
                    3270  3280# 3282  3412  3414  3420  3429  3439# 3441  3508  3510  3515  3524
                    3534# 3535  3602  3604  3610  3629  3639# 3641  3821  3823  3828  3847  3857#
                    3859  4039  4041  4047  4066  4076# 4078  4258  4260  4265  4283  4293# 4295
                    4475  4477  4484  4498  4508# 4510  4672  4674  4680  4692  4702# 4704  4809
                    4811  4816  4828  4838# 4841  4946  4948  4954  4973  4983# 4985  5142  5144
                    5149  5168  5178# 5180  5337  5339  5345  5361  5371# 5373  5501  5503  5508
                    5524  5534# 5536  5664  5666  5672  5691  5701# 5703  5864  5866  5871  5890
                    5900# 5902  6063  6065  6071  6086  6096# 6098  6359  6361  6367  6379  6389#
                    6391  6469  6471  6477  6491  6501# 6503  6688  6690  6695  6709  6719# 6721
                    6906  6908  6914  6934  6944# 6946  7092  7094  7099  7119  7129# 7131  7277
                    7279  7285  7301  7311# 7313  7430  7432  7437  7453  7463# 7465  7582  7584
                    7590  7611  7621# 7623  7764  7766  7771  7792  7802# 7804  7945  7947  7953
                    7970  7980# 7982  8110  8112  8117  8133  8143# 8145  8273  8275  8281  8306
                    8316# 8318  8445  8447  8451  8500  8525  8535# 8537  8664  8666  8670  8720
                    8733  8743# 8745  8846  8859  8869# 8871  10382
$TPB    001144      1484# 9263* 9265
$TPFLG  001151      1488# 9240  9265
$TPS    001142      1483# 9261  9265
$TRAP   061650      1893  1972  9597#
$TRP    = 000024    9604# 9614# 9615# 9616# 9617# 9618# 9619# 9620# 9621# 9622# 9623#
$TRPAD  061670      9601  9612#
$TSTNM  001102      1467# 2024* 2227* 2478* 2729* 2924* 3116* 3277* 3436* 3531* 3636* 3854* 4073*
                    4290* 4505* 4699* 4835* 4980* 5175* 5368* 5531* 5698* 5897* 6093* 6386* 6498*
                    6716* 6941* 7126* 7308* 7460* 7618* 7799* 7977* 8140* 8313* 8532* 8740* 8866*
                    9049* 9085  9120* 9122  9126  9141  9144  9161
$TTYIN  061405      9437  9438  9450  9468  9482  9487#
$TYPBN= ****** U    9618
$TYPDS  060664      9356# 9617
$TYPE   060330      9240# 9604  9613
$TYPEC  060420      9252  9259  9261# 9262
$TYPOC  060462      9296# 9614
```

DERPPA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$TYPON  060476          9295    9298*   9616
$TYPOS  060436          9291*   9615
$XTSTR  057670          9096*
$STRP = 000002          9603*   9614    9615    9616    9617    9618    9619    9620    9621    9622    9623
$OFILL  060661          9292*   9296*   9306    9341*
.     = 070650          1434*   1438    1440*   1460*   1463*   1503    1887    1899    1900    1934    1948    1949    1953
                        2056    2077    2263    2275    2379    2384    2389    2394    2399    2404    2409    2414    2419
                        2424    2429    2434    2439    2513    2525    2629    2634    2639    2644    2649    2654    2659
                        2664    2669    2674    2679    2684    2689    3196    3357    3662    3674    3880    3892    4092
                        4097    4120    4140    4309    4314    4337    4357    4524    4529    4552    4572    4623    4736
                        4748    4873    4885    5047    5242    5405    5417    5568    5580    5735    5747    5771    5775
                        5805    5934    5946    5970    5974    6004    6143    6195    6300    6992    7177    7424    7576
                        7682    7685    7702    7863    7866    7883    8011    8174    8387    8467    8481    8495    8606
                        8686    8700    8714    8785    8789    8911    8915    9047    9075    9079    9125    9126    9161
                        9217*   9265    9410*   9415    9487*   9488    9542    10271*  10320*
```

```
AA          2232#  2482
BB          2734#  2929
BB1         2734#  2807  3002
CC          3121#  3282
CC1         3441#  3535
CHECK       1078#  2040  2061  2085  2097  2113  2128  2145  2160  2759  2771  2796  2811  2954  2966
            2981   3006  3200  3217  3361  3378  3687  3704  3730  3747  3905  3922  3948  3965  4104
            4124   4161  4238  4321  4341  4378  4455  4536  4556  4635  4652  5051  5068  5246  5263
            5425   5588  5760  5959  6151  6257  6556  6575  6589  6661  6774  6793  6807  6879  6976
            7006   7018  7161  7191  7203  7671  7691  7852  7872  8016  8037  8179  8200  8456  8470
            8484   8675  8689  8703
CLRATA      1078#  4709  4846  4990  5185  5378  5541  5708  5907  6103  6396  6508  6676  6726  6894
            6951   7136  7318  7470  7628  7809  7987  8150  8323  8542  8750  8876
COMMEN         1#  1208#
DD          4704#  4841
EE          4985#  5180
ENDCOM         1#  1208#
ERROR       1108#  2052  2073  2092  2104  2125  2140  2157  2172  2185  2264  2276  2304  2318  2343
            2356   2362  2368  2380  2385  2390  2395  2400  2405  2410  2415  2420  2425  2430  2435
            2440   2514  2526  2554  2568  2593  2606  2612  2618  2630  2635  2640  2645  2650  2655
            2660   2665  2670  2675  2680  2685  2690  2766  2783  2798  2823  2836  2862  2875  2881
            2887   2961  2978  2993  3018  3031  3057  3070  3076  3082  3168  3181  3187  3193  3212
            3229   3245  3329  3342  3348  3354  3373  3390  3406  3477  3490  3496  3502  3571  3584
            3590   3596  3663  3675  3699  3716  3742  3759  3787  3802  3808  3814  3881  3893  3917
            3934   3960  3977  4005  4020  4026  4032  4116  4136  4173  4203  4218  4224  4230  4250
            4333   4353  4390  4420  4435  4441  4447  4467  4548  4568  4595  4610  4615  4620  4647
            4664   4737  4749  4777  4792  4797  4802  4874  4886  4914  4929  4934  4939  5043  5048
            5063   5080  5110  5125  5130  5135  5238  5243  5258  5275  5305  5320  5325  5330  5406
            5418   5437  5469  5484  5489  5494  5569  5581  5600  5632  5647  5652  5657  5736  5748
            5767   5772  5801  5806  5832  5847  5852  5857  5935  5947  5966  5971  6000  6005  6031
            6046   6051  6056  6137  6145  6164  6191  6196  6222  6237  6242  6247  6269  6296  6301
            6327   6342  6347  6352  6438  6453  6458  6463  6548  6568  6582  6601  6626  6641  6647
            6653   6673  6766  6786  6800  6819  6844  6859  6865  6871  6891  6988  7013  7025  7060
            7075   7080  7085  7173  7198  7210  7245  7260  7265  7270  7367  7390  7405  7410  7415
            7425   7519  7542  7557  7562  7567  7577  7678  7698  7731  7746  7751  7756  7859  7879
            7912   7927  7932  7937  8028  8049  8079  8094  8099  8104  8191  8212  8242  8257  8262
            8267   8383  8388  8414  8429  8434  8439  8463  8468  8477  8482  8491  8496  8602  8607
            8633   8648  8653  8658  8682  8687  8696  8701  8710  8715  8782  8816  8831  8837  8843
            8908   8942  8957  8963  8969
ESCAPE         1#  1208#
FF          5373#  5536
FF1         5703#  5902
GG          6098#
HEAD        1078#  2017  2220  2471  2722  2917  3109  3270  3429  3524  3629  3847  4066  4283  4498
            4692   4828  4973  5168  5361  5524  5691  5890  6086  6379  6491  6709  6934  7119  7301
            7453   7611  7792  7970  8133  8306  8525  8733  8859
HH          6391#
II          6503#  6721
JJ          6946#  7131
KK          7313#  7465
LL          8318#  8537
LL1         8318#  8451  8670
LOOP        1078#  2189  2441  2691  2889  3084  3246  3407  3504  3598  3816  4034  4253  4470  4667
            4804   4941  5137  5332  5496  5659  5859  6058  6354  6465  6683  6901  7087  7272  7426
            7578   7759  7940  8106  8269  8441  8660
MM          7982#  8145
```

```
MMO     4510#
MM1     4078#  4295
MORETA  1452#  1503
MSG     2001#  2006  2201#  2204  2452#  2455  2703#  2706  2900#  2903  3096#  3099  3257#  3260  3419#
        3422   3514#  3517  3609#  3612  3827#  3830  4046#  4049  4264#  4267  4483#  4486  4679#  4682
        4815#  4818  4953#  4956  5148#  5151  5344#  5347  5507#  5510  5671#  5674  5870#  5873  6070#
        6073   6366#  6369  6476#  6479  6694#  6697  6913#  6916  7098#  7101  7284#  7287  7436#  7439
        7589#  7592  7770#  7773  7952#  7955  8116#  8119  8280#  8283  8499#  8502  8719#  8722  8845#
        8848
MULT       1#  1208#
NEUTRA  1078#  2324  2574  2842  3037  3149  3310  3458  3552  3768  3986  4184  4401  4576  4758
        4895   5091  5286  5450  5613  5813  6012  6203  6308  6419  6607  6825  7041  7226  7371
        7523   7712  7893  8060  8223  8395  8614  8797  8923
NEWTST     1#  1208#  2004  2202  2453  2704  2901  3097  3258  3420  3515  3610  3828  4047  4265
        4484   4680  4816  4954  5149  5345  5508  5672  5871  6071  6367  6477  6695  6914  7099
        7285   7437  7590  7771  7953  8117  8281  8500  8720  8846
NN      7623#  7804
OO      3641#  3859
POP        1#  1208#  9397  9531  9581
PUSH       1#  1208#  9356  9505  9561
RELEAS  1078#  3143  3304  3762  3980  4178  4395  4752  4889  5021  5085  5216  5280  5444  5607
        5779   5807  5978  6006  6169  6197  6274  6302  7035  7220  7706  7887  8054  8217  8361
        8389   8580  8608  8791  8917
RR      8745#  8871
SCOPE   1109#  2018  2221  2472  2723  2918  3110  3271  3430  3525  3630  3848  4067  4284  4499
        4693   4829  4974  5169  5362  5525  5692  5891  6087  6380  6492  6710  6935  7120  7302
        7454   7612  7793  7971  8134  8307  8526  8734  8860  9045
SEIZE   1078#  2249  2499  3134  3295  3648  3866  4144  4361  4722  4859  5003  5198  5391  5554
        5721   5920  6520  6738  6963  7148  7331  7483  7641  7822  8334  8553
SELECT  1078#  2037  2058  2083  2095  2111  2143  2255  2266  2281  2505  2516  2531  2739  2757
        2769   2826  2934  2952  2964  3021  3146  3198  3215  3238  3307  3359  3376  3399  3446
        3540   3654  3665  3685  3702  3728  3745  3765  3872  3883  3903  3920  3946  3963  3983
        4102   4122  4181  4236  4319  4339  4398  4453  4534  4554  4633  4650  4728  4739  4755
        4865   4876  4892  5010  5024  5034  5037  5049  5066  5088  5205  5219  5229  5232  5244
        5261   5283  5397  5408  5419  5447  5560  5571  5582  5610  5727  5738  5753  5782  5792
        5795   5810  5926  5937  5952  5981  5991  5994  6009  6125  6133  6139  6172  6182  6185
        6200   6254  6277  6287  6290  6305  6538  6549  6659  6756  6767  6877  6995  7003  7016
        7038   7180  7188  7201  7223  7361  7513  7648  7655  7709  7829  7836  7890  7998  8031
        8057   8161  8194  8220  8344  8347  8353  8358  8364  8374  8377  8392  8563  8566  8572
        8577   8583  8593  8596  8611  8773  8794  8899  8920
SETATA  1078#  4084  4301  4516
SETTRA  9604#  9614  9615  9616  9617  9618  9619  9620  9621  9622
SETUP      1#  1208#  1882
SKIP       1#  1208#  2194  2196  2446  2448  2696  2698  2894  2896  3089  3091  3251  3253  3412
        3414   3508  3510  3602  3604  3821  3823  4039  4041  4258  4260  4475  4477  4672  4674
        4809   4811  4946  4948  5142  5144  5337  5339  5501  5503  5664  5666  5864  5866  6063
        6065   6359  6361  6469  6471  6688  6690  6906  6908  7092  7094  7277  7279  7430  7432
        7582   7584  7764  7766  7945  7947  8110  8112  8273  8275  8445  8447  8664  8666
SLASH      1#  1208#
SPACE   1208#
STARS      1#  1208#  1453  1539  2004  2016  2202  2219  2453  2470  2704  2721  2901  2916  3097
        3108   3258  3269  3420  3428  3515  3523  3610  3628  3828  3846  4047  4065  4265  4282
        4484   4497  4680  4691  4816  4827  4954  4972  5149  5167  5345  5360  5508  5523  5672
        5690   5871  5889  6071  6085  6367  6378  6477  6490  6695  6708  6914  6933  7099  7118
        7285   7300  7437  7452  7590  7610  7771  7791  7953  7969  8117  8132  8281  8305  8500
        8524   8720  8732  8846  8858  9034  9079  9126  9161  9218  9265  9343  9411  9428  9488
```

DERPPA.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
          9542    9588
STARSS    1078#   1209    1213    1244    1248    1404    1408    1530    1534    1876    1880    1995    1999    2030    2034
          2080    2108    2176    2233    2242    2248    2278    2297    2313    2320    2371    2483    2492    2498    2528
          2547    2563    2570    2621    2735    2742    2749    2754    2802    2808    2829    2839    2930    2937    2944
          2949    2997    3003    3024    3034    3122    3127    3133    3142    3234    3283    3288    3294    3303    3395
          3442    3450    3455    3536    3544    3549    3642    3647    3677    3682    3720    3725    3860    3865    3895
          3900    3938    3943    4079    4083    4099    4143    4152    4158    4177    4233    4296    4300    4316    4360
          4369    4375    4394    4450    4511    4515    4531    4575    4625    4630    4705    4721    4751    4842    4858
          4888    4986    5002    5013    5018    5084    5181    5197    5208    5213    5279    5374    5390    5422    5441
          5537    5553    5585    5604    5704    5720    5750    5757    5778    5903    5919    5949    5956    5977    6099
          6115    6121    6128    6148    6251    6392    6409    6416    6504    6519    6528    6533    6541    6553    6572
          6586    6606    6656    6722    6737    6746    6751    6759    6771    6790    6804    6824    6874    6947    6962
          6971    7000    7029    7034    7132    7147    7156    7185    7214    7219    7314    7330    7339    7343    7351
          7357    7370    7418    7466    7482    7491    7495    7503    7509    7522    7570    7624    7640    7651    7658
          7663    7668    7687    7705    7805    7821    7832    7839    7844    7849    7868    7886    7983    8002    8007
          8013    8034    8053    8146    8165    8170    8176    8197    8216    8319    8538    8746    8872    8973    8977
          9028    9032    9624    9628    9697    9701   10323   10327
SUBTAG    1078#   2029    2232    2482    2734    2929    3121    3282    3441    3535    3641    3859    4078    4295    4510
          4704    4841    4985    5180    5373    5536    5703    5902    6098    6391    6503    6721    6946    7131    7313
          7465    7623    7804    7982    8145    8318    8537    8745    8871
SWITCH    1078#   5027    5222    5785    5984    6175    6280    8367    8586
TIMER     1078#   2241    2491    2741    2936    3126    3287    6532    6750    7342    7350    7494    7502
TRMTRP    9604#
TYPBIN       1#   1208#
TYPDEC       1#   1208#   9060    9204
TYPNUM       1#   1208#
TYPOCS       1#   1208#
TYPOCT       1#   1208#   9176    9201
TYPTXT       1#   1208#
USEREO    9034#   9045
$$CMRE    1453#   1491
$$CMTM    1453#   1492    1493    1494    1495    1496
$$ESCA       1#   1208#
$$NEWT       1#   1208#   2004    2202    2453    2704    2901    3097    3258    3420    3515    3610    3828    4047    4265
          4484    4680    4816    4954    5149    5345    5508    5672    5871    6071    6367    6477    6695    6914    7099
          7285    7437    7590    7771    7953    8117    8281    8500    8720    8846
$$SET     9604#   9614    9615    9616    9617    9618    9619    9620    9621    9622
$$SKIP       1#   1208#   2194    2196    2446    2448    2696    2698    2894    2896    3089    3091    3251    3253    3412
          3414    3508    3510    3602    3604    3821    3823    4039    4041    4258    4260    4475    4477    4672    4674
          4809    4811    4946    4948    5142    5144    5337    5339    5501    5503    5664    5666    5864    5866    6063
          6065    6359    6361    6469    6471    6688    6690    6906    6908    7092    7094    7277    7279    7430    7432
          7582    7584    7764    7766    7945    7947    8110    8112    8273    8275    8445    8447    8664    8666
.EQUAT       1#   1078#   1103
.HEADE       1#   1078#   1079
.KT11        1#
.SETUP       1#   1078#   1876
.SWRHI       1#   1078#   1090
.SWRLO    1078#   1101#
.$CATC       1#   1078#   1431
.$CMTA       1#   1078#   1453
.$DB2D       1#
.$DB2O       1#
.$DIV        1#
.$EOP        1#   1078#   9034
.$ERRO       1#   1078#   9126
.$ERRT       1#   1078#   9161
```

.$MULT	1#		
.$POWE	1#		
.$RAND	1#		
.$RDDE	1#		
.$RDOC	1#	1078#	9488
.$READ	1#	1078#	9411
.$SAVE	1#	1078#	9542
.$SB2D	1#		
.$SB2O	1#		
.$SCOP	1#	1078#	9079
.$SIZE	1#		
.$SUPR	1#		
.$TRAP	1#	1078#	9588
.$TYPB	1#		
.$TYPD	1#	1078#	9343
.$TYPE	1#	1078#	9218
.$TYPO	1#	1078#	9265

```
ADD     1926  1985  2043  2064  2088  2100  2116  2131  2148  2163  2260  2329  2377  2382  2387
        2392  2397  2402  2407  2412  2417  2422  2427  2432  2437  2510  2579  2627  2632  2637
        2642  2647  2652  2657  2662  2667  2672  2677  2682  2687  2762  2774  2789  2814  2848
        2957  2969  2984  3009  3043  3154  3203  3220  3315  3364  3381  3463  3557  3659  3690
        3707  3733  3750  3773  3877  3908  3925  3951  3968  3991  4107  4127  4164  4189  4241
        4324  4344  4381  4406  4458  4539  4559  4581  4638  4655  4733  4753  4870  4900  5033
        5054  5071  5096  5228  5249  5266  5291  5402  5428  5455  5565  5591  5618  5732  5763
        5791  5818  5931  5962  5990  6017  6155  6181  6208  6260  6286  6313  6424  6559  6578
        6592  6612  6664  6777  6796  6810  6830  6882  6979  7009  7021  7046  7164  7194  7206
        7231  7376  7528  7674  7694  7717  7855  7875  7898  8019  8040  8065  8182  8203  8228
        8373  8400  8459  8473  8487  8592  8619  8678  8692  8706  8768  8802  8894  8928  8992
        9000  9001  9007  9024  9184  9250  9294  9304  9376  9526
ASL     1960  9181  9182  9183  9519  9521  9523
ASLB    9381
ASR     9022  9023
BCC     9382
BEQ     1981  2020  2048  2056  2069  2077  2091  2103  2121  2136  2153  2168  2184  2194  2196
        2223  2263  2275  2349  2361  2367  2379  2384  2389  2394  2399  2404  2409  2414  2419
        2424  2429  2434  2439  2446  2448  2474  2513  2525  2599  2611  2617  2629  2634  2639
        2644  2649  2654  2659  2664  2669  2674  2679  2684  2689  2696  2698  2725  2765  2779
        2794  2819  2868  2880  2886  2894  2896  2920  2960  2974  2989  3014  3063  3075  3081
        3089  3091  3112  3174  3186  3192  3196  3208  3225  3251  3253  3273  3335  3347  3353
        3357  3369  3386  3412  3414  3432  3483  3495  3501  3508  3510  3527  3577  3589  3595
        3602  3604  3632  3662  3674  3695  3712  3738  3755  3793  3807  3813  3821  3823  3850
        3880  3892  3913  3930  3956  3973  4011  4025  4031  4039  4041  4069  4092  4097  4112
        4120  4132  4140  4169  4209  4223  4229  4246  4258  4260  4286  4309  4314  4329  4337
        4349  4357  4386  4426  4440  4446  4463  4475  4477  4501  4524  4529  4544  4552  4564
        4572  4601  4614  4619  4623  4643  4660  4672  4674  4695  4736  4748  4783  4796  4801
        4809  4811  4831  4873  4885  4920  4933  4938  4946  4948  4976  5040  5042  5047  5059
        5076  5116  5129  5134  5142  5144  5171  5235  5237  5242  5254  5271  5311  5324  5329
        5337  5339  5364  5405  5417  5433  5475  5488  5493  5501  5503  5527  5568  5580  5596
        5638  5651  5656  5664  5666  5694  5735  5747  5766  5771  5775  5798  5800  5805  5838
        5851  5856  5864  5866  5893  5934  5946  5965  5970  5974  5997  5999  6004  6037  6050
        6055  6063  6065  6089  6160  6188  6190  6195  6228  6241  6246  6265  6293  6295  6300
        6333  6346  6351  6359  6361  6382  6444  6457  6462  6469  6471  6494  6564  6581  6597
        6632  6646  6652  6669  6688  6690  6712  6782  6799  6815  6850  6864  6870  6887  6906
        6908  6937  6984  7012  7024  7066  7079  7084  7092  7094  7122  7169  7197  7209  7251
        7264  7269  7277  7279  7304  7396  7409  7414  7430  7432  7456  7548  7561  7566  7582
        7584  7614  7677  7682  7697  7702  7737  7750  7755  7764  7766  7795  7858  7863  7878
        7883  7918  7931  7936  7945  7947  7973  8011  8024  8045  8085  8098  8103  8110  8112
        8136  8174  8187  8208  8248  8261  8266  8273  8275  8309  8380  8382  8387  8420  8433
        8438  8445  8447  8462  8467  8476  8481  8490  8495  8528  8599  8601  8606  8639  8652
        8657  8664  8666  8681  8686  8695  8700  8709  8714  8736  8766  8781  8785  8822  8836
        8842  8862  8892  8907  8911  8948  8962  8968  9009  9047  9064  9108  9114  9143  9146
        9186  9191  9197  9208  9321  9456  9514
BGE     7424  7576  8789  8915  9117
BGT     7422  7574  9055  9328  9390  9516
BHIS    2317  2567
BIC     1914  1916  1927  2046  2050  2067  2071  2119  2123  2134  2138  2151  2155  2166  2170
        2273  2334  2338  2359  2365  2523  2584  2588  2609  2615  2777  2781  2792  2796  2817
        2821  2853  2857  2878  2884  2972  2976  2987  2991  3012  3016  3048  3052  3073  3079
        3159  3163  3184  3190  3206  3210  3223  3227  3320  3324  3345  3351  3367  3371  3384
        3388  3468  3472  3493  3499  3562  3566  3587  3593  3672  3693  3697  3710  3714  3736
        3740  3753  3757  3778  3782  3805  3811  3890  3911  3915  3928  3932  3954  3958  3971
        3975  3996  4000  4023  4029  4110  4114  4130  4134  4167  4171  4194  4198  4221  4227
        4244  4248  4327  4331  4347  4351  4384  4388  4411  4415  4438  4444  4461  4465  4542
```

```
                4546    4562    4566    4586    4590    4641    4645    4658    4662    4746    4768    4772    4883    4905    4909
                5057    5061    5074    5078    5101    5105    5252    5256    5269    5273    5296    5300    5415    5431    5435
                5460    5464    5578    5594    5598    5623    5627    5745    5823    5827    5944    6022    6026    6158    6162
                6213    6217    6263    6267    6318    6322    6429    6433    6562    6566    6595    6599    6617    6621    6644
                6650    6667    6671    6780    6784    6813    6817    6835    6839    6862    6868    6885    6889    6982    6986
                7051    7055    7167    7171    7236    7240    7381    7385    7533    7537    7722    7726    7903    7907    8022
                8026    8043    8047    8070    8074    8185    8189    8206    8210    8233    8237    8405    8409    8624    8628
                8779    8807    8811    8834    8840    8905    8933    8937    8960    8966    9052    9318    9426    9525
BIS             1917    2051    2072    2124    2139    2156    2171    2782    2797    2822    2977    2992    3017    3211    3228
                3372    3389    3698    3715    3741    3758    3916    3933    3959    3976    4115    4135    4172    4249    4332
                4352    4389    4466    4547    4567    4646    4663    5062    5079    5257    5274    5436    5599    6163    6268
                6567    6600    6672    6785    6818    6890    6987    7172    8027    8048    8190    8211    9323    9324    9384
                9385
BISB            9173
BIT             2186    2195    2447    2697    2895    3090    3252    3413    3509    3603    3822    4040    4259    4476    4673
                4810    4947    5143    5338    5502    5665    5770    5865    5969    6064    6360    6470    6689    6907    7093
                7278    7431    7583    7765    7946    8010    8111    8173    8274    8446    8466    8480    8494    8665    8685
                8699    8713    8784    8910    9111    9145    9152
BLE             1956
BLO             9451
BLOS            1909    3439
BLT             9258    9329    9373    9389    9518
BMI             9094    9380
BNE             1887    1953    2187    2301    2303    2340    2342    2354    2551    2553    2590    2592    2604    2833    2835
                2859    2861    2873    3028    3030    3054    3056    3068    3165    3167    3179    3242    3244    3326    3328
                3340    3403    3405    3474    3476    3488    3568    3570    3582    3784    3786    3798    4002    4004    4016
                4200    4202    4214    4417    4419    4431    4592    4594    4606    4774    4776    4788    4911    4913    4925
                5107    5109    5121    5302    5304    5316    5466    5468    5480    5629    5631    5643    5829    5831    5843
                6028    6030    6042    6132    6136    6143    6219    6221    6233    6324    6326    6338    6435    6437    6449
                6545    6547    6623    6625    6637    6763    6765    6841    6843    6855    6992    7057    7059    7071    7177
                7242    7244    7256    7349    7364    7366    7387    7389    7401    7501    7516    7518    7539    7541    7553
                7685    7728    7730    7742    7866    7909    7911    7923    8076    8078    8090    8239    8241    8253    8411
                8413    8425    8630    8632    8644    8813    8815    8827    8939    8941    8953    9066    9068    9112    9153
                9174    9200    9247    9254    9319    9378    9443    9445    9461    9465    9475
BPL             2021    2224    2475    2726    2921    3113    3274    3433    3528    3633    3851    4070    4287    4502    4696
                4832    4977    5172    5365    5528    5695    5894    6090    6383    6495    6713    6938    7123    7305    7457
                7615    7796    7974    8137    8310    8529    8737    8863    9011    9157    9241    9262    9317    9364    9394
                9424
BR              1911    1934    1958    1987    1991    8991    8999    9096    9102    9105    9179    9203    9210    9243    9260
                9295    9310    9331    9375    9392    9454    9463    9469    9471    9527    9540
CLR             1885    1896    1897    1945    1946    2040    2044    2061    2065    2085    2097    2113    2128    2145    2160
                2198    2238    2239    2240    2245    2254    2261    2327    2450    2488    2489    2490    2495    2504    2511
                2577    2700    2745    2759    2763    2771    2786    2811    2846    2898    2940    2954    2958    2966    2981
                3006    3041    3093    3130    3139    3152    3200    3204    3217    3221    3255    3291    3300    3313    3361
                3365    3378    3382    3416    3461    3512    3555    3606    3653    3660    3687    3691    3704    3730    3734
                3747    3771    3825    3871    3878    3905    3909    3922    3948    3952    3965    3989    4043    4089    4094
                4104    4124    4149    4161    4165    4187    4238    4262    4306    4311    4321    4341    4366    4378    4382
                4404    4455    4479    4521    4526    4536    4556    4579    4635    4652    4676    4713    4717    4734    4761
                4813    4850    4854    4871    4898    4950    4994    4998    5008    5016    5030    5051    5055    5068    5094
                5146    5189    5193    5203    5211    5225    5246    5250    5263    5289    5341    5382    5386    5403    5425
                5429    5453    5505    5545    5549    5566    5588    5592    5616    5668    5712    5716    5726    5733    5760
                5764    5788    5816    5868    5911    5915    5925    5932    5959    5963    5987    6015    6067    6107    6111
                6152    6178    6206    6257    6283    6311    6363    6400    6404    6422    6473    6512    6516    6525    6536
                6556    6575    6589    6610    6661    6665    6680    6692    6730    6734    6743    6754    6774    6793    6807
                6828    6879    6883    6898    6910    6955    6959    6968    6976    7006    7010    7018    7044    7096    7140
                7144    7153    7161    7191    7195    7203    7229    7281    7322    7326    7336    7346    7354    7374    7434
```

```
        7474  7478  7488  7498  7506  7526  7586  7632  7636  7646  7654  7671  7675  7691  7695
        7715  7768  7813  7817  7827  7835  7852  7856  7872  7876  7896  7949  7991  7995  8016
        8037  8041  8063  8114  8154  8158  8179  8200  8204  8226  8277  8327  8331  8339  8349
        8350  8351  8370  8398  8449  8456  8460  8470  8474  8484  8488  8546  8550  8558  8568
        8569  8570  8589  8617  8668  8675  8679  8689  8693  8703  8707  8754  8758  8763  8776
        8783  8800  8880  8884  8889  8902  8909  8926  8984  9012  9049  9050  9110  9172  9308
        9367  9370  9436  9459  9511  9512
CLRB    2197  2449  2699  2897  3092  3254  3415  3511  3605  3824  4042  4261  4478  4675  4812
        4949  5145  5340  5504  5667  5867  6066  6362  6472  6691  6909  7095  7280  7433  7585
        7767  7948  8113  8276  8448  8667  9109  9396  9466  9476  9536
CMP     1886  1908  1955  2047  2068  2090  2102  2120  2135  2152  2167  2183  2262  2274  2316
        2339  2360  2366  2512  2524  2566  2589  2610  2616  2764  2778  2793  2818  2858  2879
        2885  2959  2973  2988  3013  3053  3074  3080  3164  3185  3191  3207  3224  3325  3346
        3352  3368  3385  3473  3494  3500  3567  3588  3594  3661  3673  3694  3711  3737  3754
        3783  3806  3812  3879  3891  3912  3929  3955  3972  4001  4024  4030  4111  4131  4168
        4199  4222  4228  4245  4328  4348  4385  4416  4439  4445  4462  4543  4563  4591  4613
        4618  4642  4659  4735  4747  4773  4795  4800  4872  4884  4910  4932  4937  5046  5058
        5075  5106  5128  5133  5241  5253  5270  5301  5323  5328  5404  5416  5432  5465  5487
        5492  5567  5579  5595  5628  5650  5655  5734  5746  5765  5804  5828  5850  5855  5933
        5945  5964  6003  6027  6049  6054  6159  6194  6218  6240  6245  6264  6299  6323  6345
        6350  6434  6456  6461  6563  6580  6596  6622  6645  6651  6668  6781  6798  6814  6840
        6863  6869  6886  6983  7011  7023  7056  7078  7083  7168  7196  7208  7241  7263  7268
        7386  7408  7413  7421  7423  7538  7560  7565  7573  7575  7676  7696  7727  7749  7754
        7857  7877  7908  7930  7935  8023  8044  8075  8097  8102  8186  8207  8238  8260  8265
        8386  8410  8432  8437  8461  8475  8489  8605  8629  8651  8656  8680  8694  8708  8780
        8788  8812  8835  8841  8906  8914  8938  8961  8967  9065  9067  9103  9116  9388  9438
        9450
CMPB    9253  9442  9460  9464  9474  9515  9517
COM     2053  2074  2093  2105  2126  2141  2158  2173  2271  2521  2767  2784  2799  2824  2962
        2979  2994  3019  3213  3230  3374  3391  3670  3700  3717  3743  3760  3888  3918  3935
        3961  3978  4117  4137  4174  4251  4334  4354  4391  4468  4549  4569  4648  4665  4744
        4881  5064  5081  5259  5276  5413  5438  5576  5601  5743  5768  5942  5967  6165  6270
        6413  6569  6583  6602  6674  6787  6801  6820  6892  6989  7014  7026  7174  7199  7211
        7679  7699  7860  7880  8029  8050  8192  8213  8464  8478  8492  8683  8697  8711  8771
        8897
DEC     1959  9053  9180  9449
DECB    9257  9316  9327
EMT     1108
HALT    1438  1933  2188  9158  9242
INC     1913  1962  8787  8913  9051  9115  9148  9322  9330  9374
INCB    9120  9142
IOT     1109
JMP     1442  1445  1448  1931  1954  1964  1992  2022  2199  2225  2265  2305  2344  2451  2476
        2515  2555  2594  2701  2727  2837  2863  2899  2922  3032  3058  3094  3114  3169  3197
        3232  3256  3275  3330  3358  3393  3417  3434  3478  3513  3529  3572  3607  3634  3664
        3788  3826  3852  3882  4006  4044  4071  4121  4141  4204  4263  4288  4338  4358  4421
        4480  4503  4553  4573  4596  4624  4677  4697  4738  4778  4814  4833  4875  4915  4951
        4978  5111  5147  5173  5306  5342  5366  5407  5470  5506  5529  5570  5633  5669  5696
        5737  5776  5833  5869  5895  5936  5975  6032  6068  6091  6138  6144  6146  6223  6249
        6328  6364  6384  6439  6474  6496  6627  6693  6714  6845  6911  6939  6993  6998  7061
        7097  7124  7178  7183  7246  7282  7306  7391  7435  7458  7543  7587  7616  7683  7703
        7732  7769  7797  7864  7884  7913  7950  7975  8080  8115  8138  8243  8278  8311  8415
        8450  8451  8530  8634  8669  8670  8738  8786  8790  8817  8864  8912  8916  8943  8971
        9048  9074
JSR     1930  1943  2308  2558  8346  8360  8565  8579  9070  9154  9252  9259
MOV     1883  1884  1888  1889  1890  1891  1892  1893  1894  1895  1899  1900  1903  1904  1907
```

1912	1915	1919	1922	1925	1928	1939	1941	1942	1948	1949	1952	1961	1963	1970
1971	1972	1973	1975	1982	1983	1984	1989	2002	2023	2025	2026	2027	2038	2041
2042	2045	2049	2057	2059	2062	2063	2066	2070	2078	2084	2086	2087	2089	2096
2098	2099	2101	2112	2114	2115	2117	2118	2122	2129	2130	2132	2133	2137	2144
2146	2147	2149	2150	2154	2161	2162	2164	2165	2169	2180	2182	2226	2228	2229
2230	2237	2246	2253	2256	2257	2258	2259	2267	2268	2269	2270	2272	2282	2283
2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2306	2307	2310
2311	2323	2328	2330	2332	2333	2336	2337	2345	2346	2350	2351	2355	2357	2358
2363	2364	2375	2376	2378	2381	2383	2386	2388	2391	2393	2396	2398	2401	2403
2406	2408	2411	2413	2416	2418	2421	2423	2426	2428	2431	2433	2436	2438	2477
2479	2480	2481	2487	2496	2503	2506	2507	2508	2509	2517	2518	2519	2520	2522
2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543	2544	2545	2556
2557	2560	2561	2573	2578	2580	2582	2583	2586	2587	2595	2596	2600	2601	2605
2607	2608	2613	2614	2625	2626	2628	2631	2633	2636	2638	2641	2643	2646	2648
2651	2653	2656	2658	2661	2663	2666	2668	2671	2673	2676	2678	2681	2683	2686
2688	2728	2730	2731	2732	2740	2746	2747	2752	2758	2760	2761	2770	2772	2773
2775	2776	2780	2787	2788	2790	2791	2795	2805	2806	2812	2813	2815	2816	2820
2827	2847	2849	2851	2852	2855	2856	2864	2865	2869	2870	2874	2876	2877	2882
2883	2923	2925	2926	2927	2935	2941	2942	2947	2953	2955	2956	2965	2967	2968
2970	2971	2975	2982	2983	2985	2986	2990	3000	3001	3007	3008	3010	3011	3015
3022	3042	3044	3046	3047	3050	3051	3059	3060	3064	3065	3069	3071	3072	3077
3078	3115	3117	3118	3119	3131	3138	3140	3147	3148	3153	3155	3157	3158	3161
3162	3170	3171	3175	3176	3180	3182	3183	3188	3189	3199	3201	3202	3205	3209
3216	3218	3219	3222	3226	3240	3276	3278	3279	3280	3292	3299	3301	3308	3309
3314	3316	3318	3319	3322	3323	3331	3332	3336	3337	3341	3343	3344	3349	3350
3360	3362	3363	3366	3370	3377	3379	3380	3383	3387	3401	3435	3437	3438	3439
3447	3448	3453	3462	3464	3466	3467	3470	3471	3479	3480	3484	3485	3489	3491
3492	3497	3498	3530	3532	3533	3534	3541	3542	3547	3556	3558	3560	3561	3564
3565	3573	3574	3578	3579	3583	3585	3586	3591	3592	3635	3637	3638	3639	3652
3655	3656	3657	3658	3666	3667	3668	3669	3671	3680	3686	3688	3689	3692	3696
3703	3705	3706	3708	3709	3713	3723	3729	3731	3732	3735	3739	3746	3748	3749
3751	3752	3756	3766	3767	3772	3774	3776	3777	3780	3781	3789	3790	3794	3795
3799	3800	3801	3803	3804	3809	3810	3853	3855	3856	3857	3870	3873	3874	3875
3876	3884	3885	3886	3887	3889	3898	3904	3906	3907	3910	3914	3921	3923	3924
3926	3927	3931	3941	3947	3949	3950	3953	3957	3964	3966	3967	3969	3970	3974
3984	3985	3990	3992	3994	3995	3998	3999	4007	4008	4012	4013	4017	4018	4019
4021	4022	4027	4028	4072	4074	4075	4076	4088	4090	4093	4103	4105	4106	4108
4109	4113	4123	4125	4126	4128	4129	4133	4148	4150	4155	4162	4163	4166	4170
4182	4183	4188	4190	4192	4193	4196	4197	4205	4206	4210	4211	4215	4216	4217
4219	4220	4225	4226	4237	4239	4240	4242	4243	4247	4289	4291	4292	4293	4305
4307	4310	4320	4322	4323	4325	4326	4330	4340	4342	4343	4345	4346	4350	4365
4367	4372	4379	4380	4383	4387	4399	4400	4405	4407	4409	4410	4413	4414	4422
4423	4427	4428	4432	4433	4434	4436	4437	4442	4443	4454	4456	4457	4459	4460
4464	4504	4506	4507	4508	4520	4522	4525	4535	4537	4538	4540	4541	4545	4555
4557	4558	4560	4561	4565	4580	4582	4584	4585	4588	4589	4597	4598	4602	4603
4607	4608	4609	4611	4612	4616	4617	4628	4634	4636	4637	4639	4640	4644	4651
4653	4654	4656	4657	4661	4698	4700	4701	4702	4714	4715	4718	4719	4726	4729
4730	4731	4732	4740	4741	4742	4743	4745	4756	4757	4762	4764	4766	4767	4770
4771	4779	4780	4784	4785	4789	4790	4791	4793	4794	4798	4799	4834	4836	4837
4838	4851	4852	4855	4856	4863	4866	4867	4868	4869	4877	4878	4879	4880	4882
4893	4894	4899	4901	4903	4904	4907	4908	4916	4917	4921	4922	4926	4927	4928
4930	4931	4935	4936	4979	4981	4982	4983	4995	4996	4999	5000	5007	5009	5011
5025	5026	5031	5032	5035	5036	5038	5039	5044	5045	5050	5052	5053	5056	5060
5067	5069	5070	5072	5073	5077	5089	5090	5095	5097	5099	5100	5103	5104	5112
5113	5117	5118	5122	5123	5124	5126	5127	5131	5132	5174	5176	5177	5178	5190
5191	5194	5195	5202	5204	5206	5220	5221	5226	5227	5230	5231	5233	5234	5239

5240 5245 5247 5248 5251 5255 5262 5264 5265 5267 5268 5272 5284 5285 5290
5292 5294 5295 5298 5299 5307 5308 5312 5313 5317 5318 5319 5321 5322 5326
5327 5367 5369 5370 5371 5383 5384 5387 5388 5395 5398 5399 5400 5401 5409
5410 5411 5412 5414 5420 5426 5427 5430 5434 5448 5449 5454 5456 5458 5459
5462 5463 5471 5472 5476 5477 5481 5482 5483 5485 5486 5490 5491 5530 5532
5533 5534 5546 5547 5550 5551 5558 5561 5562 5563 5564 5572 5573 5574 5575
5577 5583 5589 5590 5593 5597 5611 5612 5617 5619 5621 5622 5625 5626 5634
5635 5639 5640 5644 5645 5646 5648 5649 5653 5654 5697 5699 5700 5701 5713
5714 5717 5718 5725 5728 5729 5730 5731 5739 5740 5741 5742 5744 5754 5755
5761 5762 5769 5773 5783 5784 5789 5790 5793 5794 5796 5797 5802 5803 5811
5812 5817 5819 5821 5822 5825 5826 5834 5835 5839 5840 5844 5845 5846 5848
5849 5853 5854 5896 5898 5899 5900 5912 5913 5916 5917 5924 5927 5928 5929
5930 5938 5939 5940 5941 5943 5953 5954 5960 5961 5968 5972 5982 5983 5988
5989 5992 5993 5995 5996 6001 6002 6010 6011 6016 6018 6020 6021 6024 6025
6033 6034 6038 6039 6043 6044 6045 6047 6048 6052 6053 6092 6094 6095 6096
6108 6109 6112 6113 6119 6124 6126 6134 6141 6153 6154 6156 6157 6161 6167
6168 6173 6174 6179 6180 6183 6184 6186 6187 6192 6193 6201 6202 6207 6209
6211 6212 6215 6216 6224 6225 6229 6230 6234 6235 6236 6238 6239 6243 6244
6256 6258 6259 6261 6262 6266 6272 6273 6278 6279 6284 6285 6288 6289 6291
6292 6297 6298 6306 6307 6312 6314 6316 6317 6320 6321 6329 6330 6334 6335
6339 6340 6341 6343 6344 6348 6349 6385 6387 6388 6389 6401 6402 6405 6406
6412 6414 6423 6425 6427 6428 6431 6432 6440 6441 6445 6446 6450 6451 6452
6454 6455 6459 6460 6497 6499 6500 6501 6513 6514 6517 6518 6524 6526 6531
6537 6539 6551 6557 6558 6560 6561 6565 6576 6577 6579 6590 6591 6593 6594
6598 6611 6613 6615 6616 6619 6620 6628 6629 6633 6634 6638 6639 6640 6642
6643 6648 6649 6660 6662 6663 6666 6670 6681 6682 6715 6717 6718 6719 6731
6732 6735 6736 6742 6744 6749 6755 6757 6769 6775 6776 6778 6779 6783 6794
6795 6797 6808 6809 6811 6812 6816 6829 6831 6833 6834 6837 6838 6846 6847
6851 6852 6856 6857 6858 6860 6861 6866 6867 6878 6880 6881 6884 6888 6899
6900 6940 6942 6943 6944 6956 6957 6960 6961 6967 6969 6974 6975 6977 6978
6980 6981 6985 6994 6996 6997 7005 7007 7008 7017 7019 7020 7022 7032 7039
7040 7045 7047 7049 7050 7053 7054 7062 7063 7067 7068 7072 7073 7074 7076
7077 7081 7082 7125 7127 7128 7129 7141 7142 7145 7146 7152 7154 7159 7160
7162 7163 7165 7166 7170 7179 7181 7182 7190 7192 7193 7202 7204 7205 7207
7217 7224 7225 7230 7232 7234 7235 7238 7239 7247 7248 7252 7253 7257 7258
7259 7261 7262 7266 7267 7307 7309 7310 7311 7323 7324 7327 7328 7335 7337
7347 7355 7362 7368 7375 7377 7379 7380 7383 7384 7392 7393 7397 7398 7402
7403 7404 7406 7407 7411 7412 7459 7461 7462 7463 7475 7476 7479 7480 7487
7489 7499 7507 7514 7520 7527 7529 7531 7532 7535 7536 7544 7545 7549 7550
7554 7555 7556 7558 7559 7563 7564 7617 7619 7620 7621 7633 7634 7637 7638
7645 7647 7649 7656 7661 7666 7672 7673 7690 7692 7693 7710 7711 7716 7718
7720 7721 7724 7725 7733 7734 7738 7739 7743 7744 7745 7747 7748 7752 7753
7798 7800 7801 7802 7814 7815 7818 7819 7826 7828 7830 7837 7842 7847 7853
7854 7871 7873 7874 7891 7892 7897 7899 7901 7902 7905 7906 7914 7915 7919
7920 7924 7925 7926 7928 7929 7933 7934 7976 7978 7979 7980 7992 7993 7996
7997 7999 8000 8005 8017 8018 8020 8021 8025 8032 8038 8039 8042 8046 8058
8059 8064 8066 8068 8069 8072 8073 8081 8082 8086 8087 8091 8092 8093 8095
8096 8100 8101 8139 8141 8142 8143 8155 8156 8159 8160 8162 8163 8168 8180
8181 8183 8184 8188 8195 8201 8202 8205 8209 8221 8222 8227 8229 8231 8232
8235 8236 8244 8245 8249 8250 8254 8255 8256 8258 8259 8263 8264 8312 8314
8315 8316 8328 8329 8332 8333 8338 8340 8341 8342 8343 8345 8348 8352 8354
8355 8356 8357 8359 8365 8366 8371 8372 8375 8376 8378 8379 8384 8385 8393
8394 8399 8401 8403 8404 8407 8408 8416 8417 8421 8422 8426 8427 8428 8430
8431 8435 8436 8457 8458 8465 8469 8471 8472 8479 8483 8485 8486 8493 8497
8531 8533 8534 8535 8547 8548 8551 8552 8557 8559 8560 8561 8562 8564 8567
8571 8573 8574 8575 8576 8578 8584 8585 8590 8591 8594 8595 8597 8598 8603

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        8604  8612  8613  8618  8620  8622  8623  8626  8627  8635  8636  8640  8641  8645  8646
        8647  8649  8650  8654  8655  8676  8677  8684  8688  8690  8691  8698  8702  8704  8705
        8712  8716  8739  8741  8742  8743  8755  8756  8759  8760  8762  8767  8769  8770  8772
        8774  8775  8777  8778  8795  8796  8801  8803  8805  8806  8809  8810  8818  8819  8823
        8824  8828  8829  8830  8832  8833  8838  8839  8865  8867  8868  8869  8881  8882  8885
        8886  8888  8893  8895  8896  8898  8900  8901  8903  8904  8921  8922  8927  8929  8931
        8932  8935  8936  8944  8945  8949  8950  8954  8955  8956  8958  8959  8964  8965  8983
        8986  8987  8988  8989  8990  8993  8995  8996  8997  8998  9002  9018  9019  9020  9021
        9056  9060  9063  9098  9099  9101  9104  9118  9119  9121  9122  9123  9144  9149  9171
        9176  9185  9190  9195  9196  9198  9201  9205  9212  9213  9244  9245  9249  9255  9291
        9299  9300  9301  9307  9314  9332  9333  9334  9335  9336  9357  9358  9359  9360  9361
        9362  9363  9368  9371  9391  9397  9398  9399  9400  9401  9403  9404  9421  9422  9435
        9437  9448  9479  9480  9481  9482  9503  9504  9505  9506  9507  9509  9510  9529  9530
        9531  9532  9533  9561  9562  9563  9564  9565  9566  9567  9568  9569  9570  9577  9578
        9579  9580  9581  9582  9583  9584  9585  9586  9597  9598  9601
MOVB    1898  1929  1947  2024  2037  2058  2083  2095  2111  2143  2179  2181  2227  2252  2255
        2266  2281  2331  2335  2347  2352  2478  2502  2505  2516  2531  2581  2585  2597  2602
        2729  2739  2757  2769  2826  2850  2854  2866  2871  2924  2934  2952  2964  3021  3045
        3049  3061  3066  3116  3137  3146  3156  3160  3172  3177  3198  3215  3239  3277  3298
        3307  3317  3321  3333  3338  3359  3376  3400  3436  3446  3465  3469  3481  3486  3531
        3540  3559  3563  3575  3580  3636  3651  3654  3665  3685  3702  3728  3745  3765  3775
        3779  3791  3796  3854  3869  3872  3883  3903  3920  3946  3963  3983  3993  3997  4009
        4014  4073  4087  4095  4102  4122  4147  4156  4181  4191  4195  4207  4212  4236  4290
        4304  4312  4319  4339  4364  4373  4398  4408  4412  4424  4429  4453  4505  4519  4527
        4534  4554  4583  4587  4599  4604  4633  4650  4699  4712  4716  4725  4728  4739  4755
        4765  4769  4781  4786  4835  4849  4853  4862  4865  4876  4892  4902  4906  4918  4923
        4980  4993  4997  5006  5010  5024  5034  5037  5049  5066  5088  5098  5102  5114  5119
        5175  5188  5192  5201  5205  5219  5229  5232  5244  5261  5283  5293  5297  5309  5314
        5368  5381  5385  5394  5397  5408  5419  5447  5457  5461  5473  5478  5531  5544  5548
        5557  5560  5571  5582  5610  5620  5624  5636  5641  5698  5711  5715  5724  5727  5738
        5753  5782  5792  5795  5810  5820  5824  5836  5841  5897  5910  5914  5923  5926  5937
        5952  5981  5991  5994  6009  6019  6023  6035  6040  6093  6106  6110  6118  6125  6133
        6140  6172  6182  6185  6200  6210  6214  6226  6231  6255  6277  6287  6290  6305  6315
        6319  6331  6336  6386  6399  6403  6407  6426  6430  6442  6447  6498  6511  6515  6523
        6538  6550  6614  6618  6630  6635  6659  6679  6716  6729  6733  6741  6756  6768  6832
        6836  6848  6853  6877  6897  6941  6954  6958  6966  6995  7004  7016  7038  7048  7052
        7064  7069  7126  7139  7143  7151  7180  7189  7201  7223  7233  7237  7249  7254  7308
        7321  7325  7334  7361  7378  7382  7394  7399  7460  7473  7477  7486  7513  7530  7534
        7546  7551  7618  7631  7635  7644  7648  7655  7709  7719  7723  7735  7740  7799  7812
        7816  7825  7829  7836  7890  7900  7904  7916  7921  7977  7990  7994  7998  8031  8057
        8067  8071  8083  8088  8140  8153  8157  8161  8194  8220  8230  8234  8246  8251  8313
        8326  8330  8337  8344  8347  8353  8358  8364  8374  8377  8392  8402  8406  8418  8423
        8532  8545  8549  8556  8563  8566  8572  8577  8583  8593  8596  8611  8621  8625  8637
        8642  8740  8753  8757  8761  8764  8773  8794  8804  8808  8820  8825  8866  8879  8883
        8887  8890  8899  8920  8930  8934  8946  8951  9141  9151  9246  9263  9292  9293  9296
        9297  9298  9302  9305  9306  9325  9366  9369  9383  9386  9395  9425  9441  9446  9452
        9457  9472  9513  9600
NEG     9303  9365
NOP     1903  1904  1944  2054  2075  2094  2106  2127  2142  2159  2174  2369  2619  2768  2785
        2800  2825  2888  2963  2980  2995  3020  3083  3194  3214  3231  3355  3375  3392  3503
        3597  3701  3718  3744  3761  3815  3919  3936  3962  3979  4033  4118  4138  4175  4231
        4252  4335  4355  4392  4448  4469  4550  4570  4621  4649  4666  4803  4940  5065  5082
        5136  5260  5277  5331  5439  5495  5602  5658  5858  6057  6166  6248  6271  6353  6464
        6570  6584  6603  6654  6675  6788  6802  6821  6872  6893  6990  7015  7027  7086  7175
        7200  7212  7271  7416  7568  7680  7700  7757  7861  7881  7938  8030  8051  8105  8193
        8214  8268  8440  8659  8844  8970  9071  9072  9073
```

```
RESET   1901  1938  1969  9069
ROL     9093  9309  9311  9312  9313  9315  9520  9522  9524
RTI     9013  9124  9160  9251  9337  9405  9427  9483  9534  9571  9587
RTS     8498  8717  9003  9026  9215  9264  9602
SUB     9010  9017  9025  9150  9372
TRAP    9604  9614  9615  9616  9617  9618  9619  9620  9621  9622
TST     1980  1986  2019  2039  2055  2060  2076  2222  2300  2302  2341  2348  2353  2473  2550
        2552  2591  2598  2603  2724  2832  2834  2860  2867  2872  2919  3027  3029  3055  3062
        3067  3111  3166  3173  3178  3195  3241  3243  3272  3327  3334  3339  3356  3402  3404
        3431  3475  3482  3487  3526  3569  3576  3581  3631  3785  3792  3797  3849  4003  4010
        4015  4068  4091  4096  4119  4139  4201  4208  4213  4285  4308  4313  4336  4356  4418
        4425  4430  4500  4523  4528  4551  4571  4593  4600  4605  4622  4694  4727  4775  4782
        4787  4830  4864  4912  4919  4924  4975  5041  5108  5115  5120  5170  5236  5303  5310
        5315  5363  5396  5467  5474  5479  5526  5559  5630  5637  5642  5693  5774  5799  5830
        5837  5842  5892  5973  5998  6029  6036  6041  6088  6131  6135  6142  6189  6220  6227
        6232  6294  6325  6332  6337  6381  6436  6443  6448  6493  6544  6546  6624  6631  6636
        6711  6762  6764  6842  6849  6854  6936  6991  7058  7065  7070  7121  7176  7243  7250
        7255  7303  7348  7360  7363  7365  7388  7395  7400  7455  7500  7512  7515  7517  7540
        7547  7552  7613  7681  7684  7701  7729  7736  7741  7794  7862  7865  7882  7910  7917
        7922  7972  8077  8084  8089  8135  8240  8247  8252  8308  8381  8412  8419  8424  8527
        8600  8631  8638  8643  8735  8765  8814  8821  8826  8861  8891  8940  8947  8952  8985
        8994  9008  9046  9100  9113  9156  9207  9248  9320  9377  9387  9444  9455  9478  9528
        9535  9599
TSTB    2193  2445  2695  2893  3088  3250  3411  3507  3601  3820  4038  4257  4474  4671  4808
        4945  5141  5336  5500  5663  5863  6062  6358  6468  6687  6905  7091  7276  7429  7581
        7763  7944  8109  8272  8444  8663  9107  9199  9240  9261  9379  9393  9423
.ASCII  1500  1501  9630 10127 10150 10172 10187 10199 10210 10219 10238 10250
.ASCIZ  1499  1502  9075  9216  9486  9633  9641  9645  9648  9652  9656  9663  9666  9670  9672
        9681  9686  9703  9707  9711  9721  9727  9735  9747  9757  9762  9772  9780  9790  9797
        9806  9815  9824  9834  9843  9853  9861  9870  9877  9887  9896  9908  9917  9925  9938
        9951  9960  9968  9976  9985  9992 10003 10012 10021 10030 10038 10049 10059 10069 10081
       10091 10102 10113 10120 10133 10142 10156 10165 10178 10193 10203 10215 10225 10231 10235
       10244 10256 10262
.BLKB   9487
.BLKW   9410
.BYTE   1467  1468  1473  1474  1485  1486  1487  1488  9078  9338  9339  9340  9341  9484  9485
       10306 10308 10310 10313 10315 10316 10317 10373 10374 10375 10376 10377 10378 10379 10380
.ENABL     1  1078
.END   10384
.ENDC   1084  1098  1100  1101  1108  1194  1208  1443  1454  1462  1464  1489  1492  1497  1498
        1499  1500  1539  1540  1876  1888  1889  1891  1893  1995  1896  1897  1899  1901  2005
        2006  2016  2017  2027  2028  2041  2045  2048  2052  2053  2055  2062  2066  2069  2073
        2074  2076  2086  2090  2091  2092  2093  2095  2098  2102  2103  2104  2105  2107  2114
        2118  2121  2125  2126  2128  2129  2133  2136  2140  2141  2143  2146  2150  2153  2157
        2158  2160  2161  2165  2168  2172  2173  2175  2195  2197  2203  2204  2219  2220  2230
        2231  2247  2255  2264  2266  2268  2270  2277  2331  2356  2357  2360  2366  2447  2449
        2454  2455  2470  2471  2481  2482  2497  2505  2514  2516  2518  2520  2527  2581  2606
        2607  2610  2616  2697  2699  2705  2706  2721  2722  2732  2733  2747  2760  2764  2765
        2766  2767  2769  2772  2776  2779  2783  2784  2786  2787  2791  2794  2798  2799  2801
        2812  2816  2819  2823  2824  2826  2850  2875  2876  2879  2885  2895  2897  2902  2903
        2916  2917  2927  2928  2942  2955  2959  2960  2961  2962  2964  2967  2971  2974  2978
        2979  2981  2982  2986  2989  2993  2994  2996  3007  3011  3014  3018  3019  3021  3045
        3070  3071  3074  3080  3090  3092  3098  3099  3108  3109  3119  3120  3132  3140  3141
        3149  3156  3181  3182  3185  3191  3195  3201  3205  3208  3212  3213  3215  3218  3222
        3225  3229  3230  3232  3252  3254  3259  3260  3269  3270  3280  3281  3293  3301  3302
        3310  3317  3342  3343  3346  3352  3356  3362  3366  3369  3373  3374  3376  3379  3383
```

DERPPA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

3386	3390	3391	3393	3413	3415	3421	3422	3428	3429	3439	3440	3465	3490	3491
3494	3500	3509	3511	3516	3517	3523	3524	3534	3535	3559	3584	3585	3588	3594
3603	3605	3611	3612	3628	3629	3639	3640	3654	3663	3665	3667	3669	3676	3688
3692	3695	3699	3700	3702	3705	3709	3712	3716	3717	3719	3731	3735	3738	3742
3743	3745	3748	3752	3755	3759	3760	3762	3768	3775	3802	3803	3806	3812	3816
3822	3824	3829	3830	3846	3847	3857	3858	3872	3881	3883	3695	3887	3894	3906
3910	3913	3917	3918	3920	3923	3927	3930	3934	3935	3937	3949	3953	3956	3960
3961	3963	3966	3970	3973	3977	3978	3980	3986	3993	4020	4021	4024	4030	4034
4040	4042	4048	4049	4065	4066	4076	4077	4105	4109	4112	4116	4117	4119	4125
4129	4132	4136	4137	4139	4150	4151	4162	4166	4169	4173	4174	4176	4184	4191
4218	4219	4222	4228	4232	4239	4243	4246	4250	4251	4253	4259	4261	4266	4267
4282	4283	4293	4294	4322	4326	4329	4333	4334	4336	4342	4346	4349	4353	4354
4356	4367	4368	4379	4383	4386	4390	4391	4393	4401	4408	4435	4436	4439	4445
4449	4456	4460	4463	4467	4468	4470	4476	4478	4485	4486	4497	4498	4508	4509
4537	4541	4544	4548	4549	4551	4557	4561	4564	4568	4569	4571	4583	4610	4611
4613	4618	4636	4640	4643	4647	4648	4650	4653	4657	4660	4664	4665	4667	4673
4675	4681	4682	4691	4692	4702	4703	4720	4728	4737	4739	4741	4743	4750	4758
4765	4792	4793	4795	4800	4804	4810	4812	4817	4818	4827	4828	4838	4839	4857
4865	4874	4876	4878	4880	4887	4895	4902	4929	4930	4932	4937	4941	4947	4949
4955	4956	4972	4973	4983	4984	5001	5009	5010	5032	5044	5049	5052	5056	5059
5063	5064	5066	5069	5073	5076	5080	5081	5083	5091	5098	5125	5126	5128	5133
5137	5143	5145	5150	5151	5167	5168	5178	5179	5196	5204	5205	5227	5239	5244
5247	5251	5254	5258	5259	5261	5264	5268	5271	5275	5276	5278	5286	5293	5320
5321	5323	5328	5332	5338	5340	5346	5347	5360	5361	5371	5372	5389	5397	5406
5408	5410	5412	5419	5426	5430	5433	5437	5438	5440	5450	5457	5484	5485	5487
5492	5496	5502	5504	5509	5510	5523	5524	5534	5535	5552	5560	5569	5571	5573
5575	5582	5589	5593	5596	5600	5601	5603	5613	5620	5647	5648	5650	5655	5659
5665	5667	5673	5674	5690	5691	5701	5702	5719	5727	5736	5738	5740	5742	5749
5761	5765	5766	5767	5768	5774	5790	5802	5807	5813	5820	5847	5848	5850	5855
5859	5865	5867	5872	5873	5889	5890	5900	5901	5918	5926	5935	5937	5939	5941
5948	5960	5964	5965	5966	5967	5973	5989	6001	6006	6012	6019	6046	6047	6049
6054	6058	6064	6066	6072	6073	6085	6086	6096	6097	6114	6153	6157	6160	6164
6165	6167	6180	6192	6197	6203	6210	6237	6238	6240	6245	6249	6258	6262	6265
6269	6270	6272	6285	6297	6302	6308	6315	6342	6343	6345	6350	6354	6360	6362
6368	6369	6378	6379	6389	6390	6407	6426	6453	6454	6456	6461	6470	6472	6478
6479	6490	6491	6501	6502	6519	6526	6527	6538	6557	6561	6564	6568	6569	6571
6576	6580	6581	6582	6583	6585	6590	6594	6597	6601	6602	6604	6614	6641	6642
6645	6651	6662	6666	6669	6673	6674	6676	6683	6689	6691	6696	6697	6708	6709
6719	6720	6737	6744	6745	6756	6775	6779	6782	6786	6787	6789	6794	6798	6799
6800	6801	6803	6808	6812	6815	6819	6820	6822	6832	6859	6860	6863	6869	6880
6884	6887	6891	6892	6894	6901	6907	6909	6915	6916	6933	6934	6944	6945	6962
6969	6970	6977	6981	6984	6988	6989	6991	7007	7011	7012	7013	7014	7016	7019
7023	7024	7025	7026	7028	7041	7048	7075	7076	7078	7083	7087	7093	7095	7100
7101	7118	7119	7129	7130	7147	7154	7155	7162	7166	7169	7173	7174	7176	7192
7196	7197	7198	7199	7201	7204	7208	7209	7210	7211	7213	7226	7233	7260	7261
7263	7268	7272	7278	7200	7286	7287	7300	7301	7311	7312	7329	7337	7338	7348
7356	7378	7405	7406	7408	7413	7431	7433	7438	7439	7452	7453	7463	7464	7481
7489	7490	7500	7508	7530	7557	7558	7560	7565	7583	7585	7591	7592	7610	7611
7621	7622	7639	7647	7648	7672	7676	7677	7678	7679	7681	7692	7696	7697	7698
7699	7701	7712	7719	7746	7747	7749	7754	7758	7765	7767	7772	7773	7791	7792
7802	7803	7820	7828	7829	7853	7857	7858	7859	7860	7862	7873	7877	7878	7879
7880	7882	7893	7900	7927	7928	7930	7935	7939	7946	7948	7954	7955	7969	7970
7980	7981	7998	8017	8021	8024	8028	8029	8031	8038	8042	8045	8049	8050	8052
8060	8067	8094	8095	8097	8102	8106	8111	8113	8118	8119	8132	8133	8143	8144
8161	8180	8184	8187	8191	8192	8194	8201	8205	8208	8212	8213	8215	8223	8230
8257	8258	8260	8265	8269	8274	8276	8282	8283	8305	8306	8316	8317	8334	8340

```
                8341    8372    8384    8389    8395    8402    8429    8430    8432    8437    8441    8446    8448    8457    8461
                8462    8463    8464    8470    8471    8475    8476    8477    8478    8484    8485    8489    8490    8491    8492
                8498    8501    8502    8524    8525    8535    8536    8553    8559    8560    8591    8603    8608    8614    8621
                8648    8649    8651    8656    8660    8665    8667    8676    8680    8681    8682    8683    8689    8690    8694
                8695    8696    8697    8703    8704    8708    8709    8710    8711    8717    8721    8722    8732    8733    8743
                8744    8761    8797    8804    8831    8832    8835    8841    8845    8847    8848    8858    8859    8869    8870
                8887    8923    8930    8957    8958    8961    8967    8971    9035    9038    9040    9041    9043    9049    9055
                9058    9059    9063    9075    9078    9079    9080    9086    9089    9093    9095    9106    9107    9109    9111
                9115    9120    9122    9125    9126    9127    9133    9142    9149    9155    9156    9160    9161    9162    9180
                9218    9219    9266    9344    9412    9415    9428    9429    9437    9439    9470    9487    9488    9489    9498
                9542    9543    9589    9598    9601    9603    9613    9614    9615    9616    9617    9618    9619    9620    9621
                9622    9623
.EQUIV          1108    1109    1111    1126    1127    1156    1157    1158    1159    1160    1161    1162    1163    1164    1165
                1184    1185    1186    1187    1188    1189    1190    1191    1192    1193
.EVEN           9217   10271   10320
.IF             1080    1098    1099    1100    1101    1106    1166    1194    1438    1453    1459    1463    1489    1492    1497
                1498    1499    1503    1539    1876    1883    1888    1889    1891    1893    1895    1896    1897    1899    2004
                2006    2016    2027    2028    2041    2044    2045    2049    2052    2054    2062    2065    2066    2070    2073
                2075    2086    2089    2090    2092    2094    2098    2101    2102    2104    2106    2114    2117    2118    2122
                2125    2127    2129    2132    2133    2137    2140    2142    2146    2149    2150    2154    2157    2159    2161
                2164    2165    2169    2172    2174    2194    2196    2202    2204    2219    2230    2231    2246    2254    2255
                2263    2265    2269    2330    2356    2359    2360    2365    2366    2446    2448    2453    2455    2470    2481
                2482    2496    2504    2505    2513    2515    2519    2580    2606    2609    2610    2615    2616    2696    2698
                2704    2706    2721    2732    2733    2746    2760    2763    2764    2766    2768    2772    2775    2776    2780
                2783    2785    2787    2790    2791    2795    2798    2800    2812    2815    2816    2820    2823    2825    2849
                2875    2878    2879    2884    2885    2894    2896    2901    2903    2916    2927    2928    2941    2955    2958
                2959    2961    2963    2967    2970    2971    2975    2978    2980    2982    2985    2986    2990    2993    2995
                3007    3010    3011    3015    3018    3020    3044    3070    3073    3074    3079    3080    3089    3091    3097
                3099    3108    3119    3120    3131    3139    3140    3141    3149    3155    3181    3184    3185    3190    3191
                3195    3201    3204    3205    3209    3212    3214    3218    3221    3222    3226    3229    3231    3251    3253
                3258    3260    3269    3280    3281    3292    3300    3301    3302    3310    3316    3342    3345    3346    3351
                3352    3356    3362    3365    3366    3370    3373    3375    3379    3382    3383    3387    3390    3392    3412
                3414    3420    3422    3428    3439    3440    3464    3490    3491    3493    3494    3499    3500    3508    3510
                3515    3517    3523    3534    3535    3558    3584    3585    3587    3588    3593    3594    3602    3604    3610
                3612    3628    3639    3640    3653    3654    3662    3664    3668    3688    3691    3692    3696    3699    3701
                3705    3708    3709    3713    3716    3718    3731    3734    3735    3739    3742    3744    3748    3751    3752
                3756    3759    3761    3768    3774    3800    3802    3805    3806    3811    3812    3816    3821    3823    3828
                3830    3846    3857    3858    3871    3872    3880    3882    3886    3906    3909    3910    3914    3917    3919
                3923    3926    3927    3931    3934    3936    3949    3952    3953    3957    3960    3962    3966    3969    3970
                3974    3977    3979    3986    3992    4018    4020    4023    4024    4029    4030    4034    4039    4041    4047
                4049    4065    4076    4077    4105    4108    4109    4113    4116    4118    4125    4128    4129    4133    4136
                4138    4149    4150    4151    4162    4165    4166    4170    4173    4175    4184    4190    4216    4218    4221
                4222    4227    4228    4232    4239    4242    4243    4247    4250    4252    4258    4260    4265    4267    4282
                4293    4294    4322    4325    4326    4330    4333    4335    4342    4345    4346    4350    4353    4355    4366
                4367    4368    4379    4382    4383    4387    4390    4392    4401    4407    4433    4435    4438    4439    4444
                4445    4449    4456    4459    4460    4464    4467    4469    4475    4477    4484    4486    4497    4508    4509
                4537    4540    4541    4545    4548    4550    4557    4560    4561    4565    4568    4570    4582    4583    4608
                4610    4613    4618    4636    4639    4640    4644    4647    4649    4653    4656    4657    4661    4664    4666
                4672    4674    4680    4682    4691    4702    4703    4710    4727    4728    4736    4738    4742    4758    4764
                4790    4792    4795    4800    4804    4809    4811    4816    4818    4827    4838    4839    4847    4864    4865
                4873    4875    4879    4895    4901    4927    4929    4932    4937    4941    4946    4948    4954    4956    4972
                4983    4984    4991    5008    5009    5010    5027    5031    5043    5049    5052    5055    5056    5060    5063
                5065    5069    5072    5073    5077    5080    5082    5091    5097    5123    5125    5128    5133    5137    5142
                5144    5149    5151    5167    5178    5179    5186    5203    5204    5205    5222    5226    5238    5244    5247
                5250    5251    5255    5258    5260    5264    5267    5268    5272    5275    5277    5286    5292    5318    5320
                5323    5328    5332    5337    5339    5345    5347    5360    5371    5372    5379    5396    5397    5405    5407
```

DERPPA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

5411 5426 5429 5430 5434 5437 5439 5450 5456 5482 5484 5487 5492 5496 5501
5503 5508 5510 5523 5534 5535 5542 5559 5560 5568 5570 5574 5589 5592 5593
5597 5600 5602 5613 5619 5645 5647 5650 5655 5659 5664 5666 5672 5674 5690
5701 5702 5709 5726 5727 5735 5737 5741 5761 5764 5765 5767 5769 5785 5789
5801 5807 5813 5819 5845 5847 5850 5855 5859 5864 5866 5871 5873 5889 5900
5901 5908 5925 5926 5934 5936 5940 5960 5963 5964 5966 5968 5984 5988 6000
6006 6012 6018 6044 6046 6049 6054 6058 6063 6065 6071 6073 6085 6096 6097
6104 6153 6156 6157 6161 6164 6166 6175 6179 6191 6197 6203 6209 6235 6237
6240 6245 6249 6258 6261 6262 6266 6269 6271 6280 6284 6296 6302 6308 6314
6340 6342 6345 6350 6354 6359 6361 6367 6369 6378 6389 6390 6397 6425 6451
6453 6454 6456 6461 6469 6471 6477 6479 6490 6501 6502 6509 6525 6526 6527
6537 6557 6560 6561 6565 6568 6570 6576 6579 6580 6582 6584 6590 6593 6594
6598 6601 6603 6613 6614 6639 6641 6644 6645 6650 6651 6662 6665 6666 6670
6673 6675 6677 6688 6690 6695 6697 6708 6719 6720 6727 6743 6744 6745 6755
6775 6778 6779 6783 6786 6788 6794 6797 6798 6800 6802 6808 6811 6812 6816
6819 6821 6831 6832 6857 6859 6862 6863 6868 6869 6880 6883 6884 6888 6891
6893 6895 6906 6908 6914 6916 6933 6944 6945 6952 6968 6969 6970 6977 6980
6981 6985 6988 6990 7007 7010 7011 7013 7015 7019 7022 7023 7025 7027 7041
7047 7073 7075 7078 7083 7087 7092 7094 7099 7101 7118 7129 7130 7137 7153
7154 7155 7162 7165 7166 7170 7173 7175 7192 7195 7196 7198 7200 7204 7207
7208 7210 7212 7226 7232 7258 7260 7263 7268 7272 7277 7279 7285 7287 7300
7311 7312 7319 7336 7337 7338 7347 7355 7377 7403 7405 7406 7408 7413 7430
7432 7437 7439 7452 7463 7464 7471 7488 7489 7490 7499 7507 7529 7555 7557
7558 7560 7565 7582 7584 7590 7592 7610 7621 7622 7629 7646 7647 7648 7672
7675 7676 7678 7680 7692 7695 7696 7698 7700 7712 7718 7744 7746 7749 7754
7758 7764 7766 7771 7773 7791 7802 7803 7810 7827 7828 7829 7853 7856 7857
7859 7861 7873 7876 7877 7879 7881 7893 7899 7925 7927 7930 7935 7939 7945
7947 7953 7955 7969 7980 7981 7988 8017 8020 8021 8025 8028 8030 8038 8041
8042 8046 8049 8051 8060 8066 8092 8094 8097 8102 8106 8110 8112 8117 8119
8132 8143 8144 8151 8180 8183 8184 8188 8191 8193 8201 8204 8205 8209 8212
8214 8223 8229 8255 8257 8260 8265 8269 8273 8275 8281 8283 8305 8316 8317
8324 8339 8340 8341 8367 8371 8383 8389 8395 8401 8427 8429 8432 8437 8441
8445 8447 8457 8460 8461 8463 8465 8471 8474 8475 8477 8479 8485 8488 8489
8491 8493 8500 8502 8524 8535 8536 8543 8558 8559 8560 8586 8590 8602 8608
8614 8620 8646 8648 8651 8656 8660 8664 8666 8676 8679 8680 8682 8684 8690
8693 8694 8696 8698 8704 8707 8708 8710 8712 8720 8722 8732 8743 8744 8751
8797 8803 8829 8831 8834 8835 8840 8841 8845 8846 8848 8858 8869 8870 8877
8923 8929 8955 8957 8960 8961 8966 8967 8971 9034 9038 9039 9040 9041 9042
9043 9045 9054 9057 9059 9063 9074 9075 9079 9085 9089 9093 9105 9107 9108
9109 9111 9113 9122 9124 9125 9126 9132 9141 9145 9152 9154 9155 9156 9159
9160 9161 9179 9195 9218 9265 9343 9411 9415 9428 9436 9438 9443 9486 9487
9488 9494 9510 9542 9588 9597 9601 9603 9604 9614 9615 9616 9617 9618 9619
9620 9621 9622

.IFF 1098 1100 1101 1106 1454 1463 1489 1540 1888 2004 2005 2006 2017 2028 2045
2048 2053 2054 2066 2069 2074 2075 2089 2090 2093 2094 2101 2102 2105 2106
2117 2121 2126 2127 2132 2136 2141 2142 2149 2153 2158 2159 2164 2168 2173
2174 2195 2197 2202 2203 2204 2220 2230 2246 2254 2255 2258 2264 2268 2269
2330 2356 2360 2366 2447 2449 2453 2454 2455 2471 2481 2496 2504 2505 2514
2516 2519 2580 2606 2610 2616 2697 2699 2704 2705 2706 2722 2733 2746 2764
2767 2768 2775 2779 2783 2785 2790 2794 2798 2800 2815 2819 2824 2825 2849
2875 2879 2885 2895 2897 2901 2902 2903 2917 2928 2941 2959 2962 2963 2970
2974 2978 2980 2985 2989 2993 2995 3010 3014 3019 3020 3044 3045 3070 3071
3073 3079 3090 3092 3097 3098 3099 3109 3119 3131 3139 3141 3149 3155 3156
3181 3182 3184 3190 3195 3205 3208 3213 3214 3222 3225 3230 3231 3252 3254
3258 3259 3260 3270 3280 3292 3300 3301 3302 3310 3316 3317 3342 3343 3345
3351 3356 3366 3369 3374 3375 3383 3386 3391 3392 3413 3415 3420 3421 3422

3429	3439	3464	3465	3491	3493	3499	3509	3511	3515	3516	3517	3524	3534	3558
3559	3585	3587	3593	3603	3605	3610	3611	3612	3629	3639	3653	3654	3657	3663
3667	3669	3692	3695	3700	3701	3708	3712	3717	3718	3735	3738	3743	3744	3751
3755	3760	3761	3768	3774	3775	3802	3803	3805	3811	3816	3822	3824	3828	3829
3830	3847	3857	3871	3872	3881	3883	3887	3910	3913	3918	3919	3926	3930	3935
3936	3953	3956	3961	3962	3969	3973	3978	3979	3986	3992	3993	4020	4021	4023
4029	4034	4040	4042	4047	4048	4049	4066	4076	4108	4112	4116	4118	4128	4132
4136	4138	4149	4151	4166	4169	4174	4175	4184	4190	4191	4218	4219	4221	4227
4232	4242	4246	4251	4252	4259	4261	4265	4266	4267	4283	4293	4325	4329	4333
4335	4345	4349	4353	4355	4366	4367	4368	4383	4386	4391	4392	4401	4407	4408
4435	4436	4438	4444	4449	4459	4463	4468	4469	4476	4478	4484	4485	4486	4498
4508	4540	4544	4548	4550	4560	4564	4568	4570	4583	4610	4611	4613	4618	4639
4643	4648	4649	4656	4660	4665	4666	4673	4675	4680	4681	4682	4692	4702	4710
4712	4716	4727	4728	4731	4737	4741	4743	4758	4764	4765	4792	4793	4795	4800
4804	4810	4812	4816	4817	4818	4828	4838	4847	4849	4853	4864	4865	4874	4876
4880	4895	4901	4902	4929	4930	4932	4937	4941	4947	4949	4954	4955	4956	4973
4983	4991	4993	4997	5008	5009	5010	5027	5032	5043	5049	5056	5059	5064	5065
5072	5076	5081	5082	5091	5097	5098	5125	5126	5128	5133	5137	5143	5145	5149
5150	5151	5168	5178	5186	5188	5192	5203	5205	5227	5238	5244	5251	5254	5259
5260	5267	5271	5276	5277	5286	5292	5293	5320	5321	5323	5328	5332	5338	5340
5345	5346	5347	5361	5371	5379	5381	5385	5396	5397	5406	5408	5412	5430	5433
5437	5439	5450	5456	5457	5484	5485	5487	5492	5496	5502	5504	5508	5509	5510
5524	5534	5542	5544	5548	5559	5560	5563	5569	5573	5575	5593	5596	5600	5602
5613	5619	5620	5647	5648	5650	5655	5659	5665	5667	5672	5673	5674	5691	5701
5709	5711	5715	5726	5727	5736	5738	5742	5765	5767	5774	5785	5790	5801	5807
5813	5819	5820	5847	5848	5850	5855	5859	5865	5867	5871	5872	5873	5890	5900
5908	5910	5914	5925	5926	5929	5935	5939	5941	5964	5966	5973	5989	6000	6006
6012	6018	6019	6046	6047	6049	6054	6058	6064	6066	6071	6072	6073	6086	6096
6104	6106	6110	6156	6160	6164	6166	6175	6180	6192	6197	6203	6209	6210	6237
6238	6240	6245	6249	6261	6265	6269	6271	6285	6297	6302	6308	6314	6315	6342
6343	6345	6350	6354	6360	6362	6367	6368	6369	6379	6389	6397	6399	6403	6425
6426	6454	6456	6461	6470	6472	6477	6478	6479	6491	6501	6509	6511	6515	6525
6527	6537	6560	6564	6569	6570	6579	6580	6582	6584	6593	6597	6602	6603	6614
6641	6642	6644	6650	6666	6669	6674	6675	6679	6680	6683	6689	6691	6695	6696
6697	6709	6719	6727	6729	6733	6743	6744	6745	6755	6778	6782	6787	6788	6797
6798	6800	6802	6811	6815	6820	6821	6832	6859	6860	6862	6868	6884	6887	6892
6893	6897	6898	6901	6907	6909	6914	6915	6916	6934	6944	6952	6954	6958	6968
6970	6980	6984	6989	6990	7011	7014	7015	7022	7023	7025	7027	7041	7047	7048
7075	7076	7078	7083	7087	7093	7095	7099	7100	7101	7119	7129	7137	7139	7143
7153	7154	7155	7165	7169	7174	7175	7196	7199	7200	7207	7208	7210	7212	7226
7232	7233	7260	7261	7263	7268	7272	7278	7280	7285	7286	7287	7301	7311	7319
7321	7325	7336	7338	7348	7355	7377	7378	7406	7408	7413	7431	7433	7437	7438
7439	7453	7463	7471	7473	7477	7488	7489	7490	7500	7507	7529	7530	7558	7560
7565	7583	7585	7590	7591	7592	7611	7621	7629	7631	7635	7646	7647	7648	7676
7679	7680	7696	7699	7700	7712	7718	7719	7746	7747	7749	7754	7758	7765	7767
7771	7772	7773	7792	7802	7810	7812	7816	7827	7829	7857	7860	7861	7877	7880
7881	7893	7899	7900	7927	7928	7930	7935	7939	7946	7948	7953	7954	7955	7970
7980	7988	7990	7994	8020	8024	8029	8030	8042	8045	8050	8051	8060	8066	8067
8094	8095	8097	8102	8106	8111	8113	8117	8118	8119	8133	8143	8151	8153	8157
8183	8187	8192	8193	8205	8208	8213	8214	8223	8229	8230	8257	8258	8260	8265
8269	8274	8276	8281	8282	8283	8306	8316	8324	8326	8330	8339	8341	8372	8383
8389	8395	8401	8402	8429	8430	8432	8437	8441	8446	8448	8461	8464	8470	8475
8478	8484	8489	8492	8498	8500	8501	8502	8525	8535	8543	8545	8549	8558	8559
8560	8586	8591	8602	8608	8614	8620	8621	8648	8649	8651	8656	8660	8665	8667
8680	8683	8689	8694	8697	8703	8708	8711	8717	8720	8721	8722	8733	8743	8751
8753	8757	8797	8803	8804	8831	8832	8834	8840	8845	8846	8847	8848	8859	8869

```
        8877  8879  8883  8923  8929  8930  8957  8958  8960  8966  8971  9035  9042  9045  9054
        9057  9075  9080  9105  9107  9109  9125  9127  9132  9145  9159  9160  9161  9162  9179
        9195  9219  9266  9344  9412  9421  9428  9429  9438  9470  9487  9489  9543  9589  9598
.IFT    2266  2516  3141  3302  3665  3883  4151  4368  4712  4739  4849  4876  4993  5010  5188
        5205  5381  5408  5544  5571  5711  5738  5910  5937  6106  6399  6511  6527  6679  6729
        6745  6897  6954  6970  7139  7155  7321  7338  7473  7490  7631  7648  7812  7829  7990
        8153  8326  8341  8545  8560  8753  8879  9111  9155  9421  9515  9535  9542
.IFTF   2258  2508  3141  3302  3657  3875  4151  4368  4713  4731  4850  4868  4994  5010  5189
        5205  5382  5400  5545  5563  5712  5730  5911  5929  6107  6400  6512  6527  6680  6730
        6745  6898  6955  6970  7140  7155  7322  7338  7474  7490  7632  7648  7813  7829  7991
        8154  8327  8341  8546  8560  8754  8880  9109  9154  9415  9511  9519  9541
.IIF    1079  1084  1089  1095  1096  1097  1098  1438  1503  1889  1891  1895  1896  1897  1899
        1900  9040  9049  9050  9061  9075  9079  9086  9087  9088  9089  9110  9122  9125  9126
        9133  9134  9135  9136  9137  9161  9177  9202  9206  9265  9415  9478  9488  9542  9613
        9614  9615  9616  9617  9618  9619  9620  9621  9622
.IRP    1503  1876  2004  2037  2083  2111  2202  2283  2291  2376  2411  2453  2533  2541  2626
        2661  2704  2901  3097  3258  3420  3515  3610  3828  4047  4102  4265  4319  4484  4534
        4680  4816  4954  5149  5345  5508  5672  5871  6071  6367  6477  6695  6914  7099  7285
        7437  7590  7771  7953  8117  8261  8341  8349  8355  8456  8500  8560  8568  8574  8675
        8720  8846  9045  9141  9357  9397  9505  9531  9561  9581 10332 10352 10362
.LIST      1  1078  1208  1438  1489  1491  1492  1493  1494  1495  1496  1497  1876  2004  2027
        2202  2230  2453  2481  2704  2732  2901  2927  3002  3097  3119  3258  3280  3420  3439
        3515  3534  3609  3610  3639  3828  3857  4047  4076  4265  4293  4482  4484  4508  4680
        4702  4816  4838  4954  4983  5018  5084  5149  5178  5213  5279  5345  5371  5508  5534
        5672  5701  5871  5900  6071  6096  6367  6389  6477  6501  6607  6655  6695  6719  6825
        6873  6914  6944  7099  7129  7285  7311  7437  7463  7590  7621  7771  7802  7953  7980
        8117  8143  8281  8316  8500  8535  8720  8743  8791  8845  8846  8869  8917  8971  9079
        9089  9428  9603  9604  9613  9614  9615  9616  9617  9618  9619  9620  9621  9622  9623
.MACRO     1  1078  1101  1452  1453  2001  2201  2232  2452  2703  2734  2900  3096  3121  3257
        3419  3441  3514  3609  3641  3827  4046  4078  4264  4483  4510  4679  4704  4815  4953
        4985  5148  5344  5373  5507  5671  5703  5870  6070  6098  6366  6391  6476  6503  6694
        6913  6946  7098  7284  7313  7436  7589  7623  7770  7952  7982  8116  8280  8318  8499
        8719  8745  8845  9034  9604
.MCALL  1078  1208
.NLIST     1  1078  1208  1438  1489  1491  1492  1493  1494  1495  1496  1497  1876  2004  2027
        2202  2230  2453  2481  2704  2732  2901  2927  3002  3097  3119  3258  3280  3420  3439
        3515  3534  3609  3610  3639  3828  3857  4047  4076  4265  4293  4482  4484  4508  4680
        4702  4816  4838  4954  4983  5018  5084  5149  5178  5213  5279  5345  5371  5508  5534
        5672  5701  5871  5900  6071  6096  6367  6389  6477  6501  6607  6655  6695  6719  6825
        6873  6914  6944  7099  7129  7285  7311  7437  7463  7590  7621  7771  7802  7953  7980
        8117  8143  8281  8316  8500  8535  8720  8743  8791  8845  8846  8869  8917  8971  9079
        9089  9428  9603  9604  9613  9614  9615  9616  9617  9618  9619  9620  9621  9622  9623
.PAGE   1453  1539
.REM       1
.REPT   1438  1491  1492
.SBTTL  1091  1104  1211  1246  1406  1432  1439  1455  1532  1541  1878  1997  2004  2202  2453
        2704  2901  3097  3258  3420  3515  3610  3828  4047  4265  4484  4680  4816  4954  5149
        5345  5508  5672  5871  6071  6367  6477  6695  6914  7099  7285  7437  7590  7771  7953
        8117  8281  8500  8720  8846  8975  9030  9036  9081  9128  9163  9220  9267  9345  9413
        9490  9544  9590  9605  9626  9699 10325
.TITLE  1079
.WORD   1438  1466  1469  1470  1471  1472  1475  1476  1477  1478  1479  1480  1489  1491  1492
        1493  1494  1495  1496  1503  1504  1505  1506  1507  1508  1509  1510  1511  1512  1513
        1514  1515  1516  1517  1518  1519  1520  1521  1522  1523  1524  1525  1526  1527  1528
        1536  1537  1965  2309  2559  9054  9057  9188  9193  9342  9538  9541 10273 10275 10277
       10280 10282 10285 10288 10290 10292 10294 10296 10298 10299 10301 10303 10331 10332 10333
```

```
      10334   10335   10336   10337   10338   10339   10340   10341   10342   10343   10344   10345   10346   10347   10348
      10349   10350   10351   10352   10353   10354   10355   10356   10357   10358   10359   10360   10361   10362   10363
      10364   10365   10366   10367   10368   10369   10382

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DERPPA.SEQ/SOL/CRF/PAGNUM/NL:TOC/NL:MC=DERPPA.SML,DERPPA.P11
RUN-TIME:  83 127 20 SECONDS
RUN-TIME RATIO: 614/232=2.6
CORE USED:  31K  (61 PAGES)
```

K03

Spooler runtime 41 Seconds, 169 KCS, 1215 disk reads, 0 disk writes, 242 pages