RM80 DU PORT TST PT 1 AH-T119A-MC JUL 1982
CZRNHAO FICHE 1 OF 2

RM80

DU PORT TST PT 1
CZRNHAO

AH-T119A-MC
FICHE 2 OF 2

JUL 1982

MADE IN USA

```
.REM    ^
```

IDENTIFICATION

PRODUCT CODE: AC-T118A-MC

PRODUCT NAME: CZRNHAO RM80 DUAL PORT TEST, PT 1

PRODUCT DATE: APRIL 1, 1982

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

CONTENTS

1. ABSTRACT

THE RM80 DUAL PORT LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RM80 DUAL PORT LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL PORT MODE IS NOT TESTED BY THIS PROGRAM.

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/70 PROCESSOR
20K MEMORY
KW11-L OR KW11-P CLOCK
TERMINAL
RH70 CONTROLLER
1 - RM80 DISK DRIVE
RM DUAL PORT TEST CABLE (P/N: 7010507-02)

2.2 PREREQUISITE PROGRAMS

RM80 DISKLESS TEST, PART 1 & 2

RM80 FUNCTIONAL TEST, PART 1, 2 & 3

THE PRELIMINARY PROGRAMS MUST BE RUN TWICE: ONCE FROM EACH PORT (A & B).

2.3 OTHER PROGRAMS

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

B. DYNAMIC OPERATION OF THE DUAL PORT OPTION IS TESTED BY THE RM80 PERFORMANCE EXERCISER PROGRAM.

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' LOADER. THE PROGRAM MAY NOT

BE INCLUDED IN AN 'XXDP' CHAIN.

4. STARTING PROCEDURES

4.1 STARTING ADDRESSES

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

B. THE RESTART ADDRESS IS LOCATION 204 (8). THE PROGRAM WILL USE THE CURRENT DRIVE ADDRESS.

C. THE PROGRAM CAN BE STARTED AT LOCATION 210 (8) TO ALLOW THE ADDRESS OF THE RH70 TO BE CHANGED.

4.2 UNIBUS & VECTOR ADDRESSES

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

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

4.3 OPERATOR ACTION

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

5. OPERATING PROCEDURES

5.1 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RM80 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

```
'SWR = NNNNNN    NEW ='
```

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

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

5.2 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

5.3 TEST SELECTION

INDIVIDUAL TESTS ARE SELECTED IN RESPONSE TO THE 'ENTER TEST NUMBER:' MESSAGE. ANY VALID TEST NUMBER CAN BE ENTERED. EACH ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN (CR). THE LOOP ON TEST SWITCH, SW<14>, MUST BE SET TO ALLOW CONTINUOUS EXECUTION OF THE SELECTED TEST.

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

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

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

5.4 TEST CABLE CONNECTION

TO TEST THE RM80 DUAL PORT OPTION WITH THIS PROGRAM,
A SPECIAL TEST CABLE MUST BE USED. (THE TEST CABLE IS
P/N: 7010507-02). THE TEST CABLE CONNECTS MASSBUS A & MASSBUS
B TOGETHER AT THE DRIVE BEING TESTED AND IS CONSTRUCTED SO
THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE DRIVE CABLE CONNECTED TO THE RM80 UNDER TEST,
THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT
OF THE DRIVE WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS.
THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S
ADDRESS PLUG.

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

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

THE TEST CABLE CONNECTION TO THE DRIVE UNDER TEST WILL
DEPEND ON WHICH PROCESSOR, RH CONTROLLER IS TO TEST THE DRIVE.
IF THE DRIVE IS TO BE TESTED BY THE PROCESSOR ON PORT A,
CONNECT THE MASSBUS CABLE FROM THE RH CONTROLLER TO J3 OF THE
RM ADPTER BACK PANEL, THEN CONNECT THE TEST CABLE (P/N: 7010507-02)
FROM J2 TO J7 OF THE BACK PANEL AND TERMINATE THE PORT 'B' AT J6.

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

6. ERRORS

WHEN THE PROGRAM ENCOUNTERS AN ERROR, THE ERROR ROUTINE IS

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

A. AN ERROR MESSAGE
B. A DATA HEADER LINE
C. A DATA LINE CONTAINING:
   1. THE TEST NUMBER
   2. THE PC (PROGRAM COUNTER VALUE) WHERE THE ERROR CALL WAS MADE
   3. CONTENTS OF THE 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 DRIVE UNDER TEST MUST HAVE THE DUAL PORT TEST CABLE CONNECTED.

7.2 LIMITATIONS

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

7.3 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 25 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 2.3 MINUTES.

7.4 REQUIRED TESTS

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

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

7.5 DISK SURFACE USAGE

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

7.6 LOOP ON ERROR OPTION

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

BECAUSE THE PROGRAM MUST RESET THE RM80 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR WAS DETECTED.

8. TEST DESCRIPTIONS

8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

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

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

8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

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

8.3 METHOD USED TO VERIFY PORT REQUESTS

THE PORT REQUEST FLOPS IN THE MAINTENANCE REGISTER ARE TESTED TO DETERMINE IF :

- A DRIVE IS IN NEUTRAL,I.E., RQA AND RQB ARE ZERO;
- A DRIVE IS SEIZED, I.E., RQA OR RQB IS ONE;
- A PORT REQUEST IS SET WHILE THE DRIVE IS SEIZED TO THE ALTERNATE PORT, I.E., RQA AND RQB ARE ONE.

TEST 1 NEUTRAL ACCESS TEST

VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS

A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RM80, THAT THE DRIVE IS ONLINE (RMDS HAS 'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.

```
                    B.  THE TEST IS REPEATED THROUGH BOTH PORTS


                  TEST 2  PORT 'A' SEIZE/TIMEOUT TEST

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

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

                    B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B';
                          VERIFY THAT 0'S ARE READ FROM EACH REGISTER.

                    C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
                          MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
                          VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
                          NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.


                  TEST 3  PORT 'B' SEIZE/TIMEOUT TEST

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

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

                    B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'A';
                          VERIFY THAT 0'S ARE READ FROM EACH REGISTER.

                    C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
                          MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
                          VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
                          NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS >500 MS.


                  TEST 4  PORT 'A' SEIZE/RELEASE TEST

                  TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

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

                    B.  SET VOLUME VALID AND CLEAR ANY ERROR

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


                  TEST 5  PORT 'B' SEIZE/RELEASE TEST

                  TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

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

                    B.  SET VOLUME VALID AND CLEAR ANY ERROR
```

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

TEST 6 PORT 'A' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 7 PORT 'B' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

TEST 10 PORT 'A' RELEASE INTERFERENCE TEST

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

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

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

C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.

D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'.

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

TEST 11 PORT 'B' RELEASE INTERFERENCE TEST

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

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

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

C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.

D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'.

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

```
                    TEST 12 PORT 'A' RELEASE W/ERRORS TEST

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

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

                      B.  WRITE 1'S INTO RMER1 THROUGH PORT 'A'.

                      C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE 'GO'
                            BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
                            THAT RMER1 HAS NOT BEEN CLEARED.

                      D.  CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.

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


                    TEST 13 PORT 'B' RELEASE W/ERRORS TEST

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

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

                      B.  WRITE 1'S INTO RMER1 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 RMER1 HAS NOT BEEN CLEARED.

                      D.  CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.

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


                    TEST 14 PORT 'A' SEIZE AND CLEAR TEST

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

                      A.  SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'A'.
                            VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

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

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

```
TEST 15 PORT 'B' SEIZE AND CLEAR TEST

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

  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'B'.
        VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

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

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


TEST 16 SEIZE 'A' BY RMCS1 TEST

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

  A.  READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'A'; VERIFY THAT
        THE DRIVE IS SEIZED.

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


TEST 17 SEIZE 'B' BY RMCS1 TEST

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

  A.  READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT
        THE DRIVE IS SEIZED.

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


TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST

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

  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1.  VERIFY THAT
        THE DRIVE HAS BEEN SEIZED.

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

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

TEST 21 PORT 'B' INHIBIT SEIZE BY RMCS1 TEST

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

A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

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

TEST 22 SEIZE BY RMAS TEST

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

A. WRITE THE APPROPRIATE DRIVE BIT INTO RMAS; VERIFY THAT THE DRIVE IS SEIZED.

B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.

TEST 23 INHIBIT SEIZE BY RMAS TEST

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

A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.

B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.

TEST 24 SET PORT 'A' REQUEST TEST

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

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

B. WRITE 0'S INTO RMDS FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.

C. ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'A' AND IS NOT SET FOR PORT 'B'.

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

```
                    TEST 25 SET PORT 'B' REQUEST TEST

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

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

                      B.  WRITE 0'S INTO RMDS FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
                            SEIZED BY PORT 'A'.

                      C.  ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
                            SWITCHED TO PORT 'B'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
                            PORT 'B' AND IS NOT SET FOR PORT 'A'.

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


                    TEST 26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR

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

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

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

                      C.  ISSUE A DRIVE CLEAR COMMAND.

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


                    TEST 27 TEST RESET ATTENTION 'B' BY DRIVE CLEAR

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

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

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

                      C.  ISSUE A DRIVE CLEAR COMMAND.

                      D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE ATTENTION

                            BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
                            'A' IS STILL SET.


                    TEST 30 RESET ATTENTION 'A' BY GO TEST

                    VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
```

```
                    SEIZING PORT.

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

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

              C.  ISSUE A NOP COMMAND.

              D.  RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE
                    ATTENTION BIT FOR PORT 'A' IS RESET, AND THE
                    ATTENTION BIT FOR PORT 'B' IS STILL SET.


            TEST 31 RESET ATTENTION 'B' BY GO TEST

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

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

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

              C.  ISSUE A NOP COMMAND.

              D.  RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE
                    ATTENTION BIT FOR PORT 'B' IS RESET, AND THE
                    ATTENTION BIT FOR PORT 'A' IS STIL SET.


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

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

              A.  SET THE ATTENTION BITS FOR BOTH PORTS.

              B.  VERIFY THAT THE DRIVE IS IN NEUTRAL.

              C.  ISSUE A MASSBUS INIT.  VERIFY THAT BOTH ATTENTION BITS HAVE
                    RESET.


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

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

              A.  SET THE ATTENTION BITS FOR BOTH PORTS.

              B.  VERIFY THE DRIVE IS IN NEUTRAL.

              C.  WRITE THE DRIVE'S ATTENTION BIT IN RMAS. VERIFY
                    THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMAS.
```

```
TEST 34 PORT 'A' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

  A.  SET THE ATTENTION BIT FOR PORT 'A'.

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

  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
        FOR THE DRIVE IS SET.


TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

  A.  SET THE ATTENTION BIT FOR PORT 'B'.

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

  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
        FOR THE DRIVE IS SET.


TEST 36 SET ATTENTION 'A' BY COMMMAND TEST

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

  A.  ISSUE A OFFSET COMMAND THROUGH PORT 'A'.

  B.  WAIT FOR THE OFFSET COMMAND TO COMPLETE ('DRY' TO BECOME
        '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
        THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.

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


TEST 37 SET ATTENTION 'B' BY COMMMAND TEST

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

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

  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
        '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND
        THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.

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


TEST 40 PORT 'A' SET VOLUME VALID TEST
```

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

                    VERIFY THAT A CHANGE IN "MUR", IN RMMR1, SETS THE ATTENTION
                          FOR PORT'A'.

                      A.  WITH PORT 'A' SELECTED, RESET AND SET "MUR" IN RMMR1,
                            USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
                            IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
                            ATTENTION IS SET.

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

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

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


                    TEST 41 PORT 'B' SET VOLUME VALID TEST

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

                    VERIFY THAT A CHANGE IN "MUR", IN RMMR1, SETS THE ATTENTION
                          FOR PORT'B'.

                      A.  WITH PORT 'B' SELECTED, RESET AND SET "MUR" IN RMMR1,
                            USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
                            IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
                            ATTENTION IS SET.

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

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

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


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

                    VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

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

                      B.  WRITE 1'S INTO RMER1 THROUGH PORT 'A' TO FORCE AN ATTENTION.
```

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


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

                  VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

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

                    B.  WRIT. 1'S INTO RMER1 THROUGH PORT 'B'.

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


                  TEST 44 PORT 'A' RETRIGGER BY DEMAND TEST

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

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

                    B.  WAIT 500 MS AND READ RMDS THROUGH PO

                    C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
                          TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)

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


                  TEST 45 PORT 'B' RETRIGGER BY DEMAND TEST

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

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

                    B.  WAIT 5C0 MS AND WRITE 0'B INTO RMDS THROUGH PORT 'A'.

                    C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
                          TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)

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


                  TEST 46 PORT 'A'  TIMEOUT/RELEASE TEST

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

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

                    B.  SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'A'.
```

C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.

D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

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

TEST 47 PORT 'B' TIMEOUT/RELEASE TEST

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

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

B. SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'B'.

C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.

D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

TEST 50 PORT 'A' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

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

B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.

C. READ RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.

D. CLEAR RMER1, RMER2 THROUGH PORT 'A'.

E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.

F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.

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

TEST 51 PORT 'B' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

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

B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.

C. READ RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.

D. CLEAR RMER1, RMER2 THROUGH PORT 'B'.

E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.

F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.

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

```
  1                                 ;*LAST REVISION 04-AUG-81
591
592                                 .TITLE  CZRNHAO RM80 DUAL PORT PT1
                                    ;*COPYRIGHT (C) 1982
                                    ;*DIGITAL EQUIPMENT CORPORATION
                                    ;*COLORADO SPGS., CO. 80919
                                    ;*
                                    ;*PROGRAM BY MIKE LEAVITT
                                    ;*
                                    ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                    ;*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81
                                    ;*
593                                 .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
594
595                                 .SBTTL  BASIC DEFINITIONS

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

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

                                    ;*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
        000006                      SP      = %6            ;;STACK POINTER
        000007                      PC      = %7            ;;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
000200                  PR4     = 200           ;;PRIORITY LEVEL 4
000240                  PR5     = 240           ;;PRIORITY LEVEL 5
000300                  PR6     = 300           ;;PRIORITY LEVEL 6
000340                  PR7     = 340           ;;PRIORITY LEVEL 7

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

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

```
000020                  BIT4=BIT04
000010                  BIT3=BIT03
000004                  BIT2=BIT02
000002                  BIT1=BIT01
000001                  BIT0=BIT00

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

                        .SBTTL  RH CONTROLLER REGISTERS

                        ;CONTROL AND STATUS REGISTER 1 (RMCS1)

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

                        ;CONTROL AND STATUS REGISTER 2 (RMCS2)

000001                  U0      = 1                     ;UNIT SELECT (BIT #0)
000002                  U1      = 2                     ;UNIT SELECT (BIT #1)
000004                  U3      = 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                  MDPE    = 400                   ;MASS BUS PARITY ERROR (BIT #8)
001000                  MXF     = 1000                  ;MISSED TRANSFER ERROR (BIT #9)
002000                  PGE     = 2000                  ;PROGRAM ERROR (BIT #10)
004000                  NEM     = 4000                  ;NON EXISTENT MEMORY (BIT #11)
010000                  NED     = 10000                 ;NON EXISTENT DRIVE (BIT #12)
020000                  UPE     = 20000                 ;UNIBUS PARITY ERROR (BIT #13)
040000                  WCE     = 40000                 ;WRITE CHECK ERROR (BIT #14)
100000                  DLT     = 100000                ;DATA LATE (BIT #15)

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

                        .SBTTL  RM REGISTERS
```

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

000001                          GO      = 1                     ;GO BIT (BIT #0)
000002                          F0      = 2                     ;FUNCTION CODE BIT #1
000004                          F1      = 4                     ;FUNCTION CODE BIT #2
000010                          F2      = 10                    ;FUNCTION CODE BIT #3
000020                          F3      = 20                    ;FUNCTION CODE BIT #4
000040                          F4      = 40                    ;FUNCTION CODE BIT #5
004000                          DVA     = 4000                  ;DEVICE AVAILABLE (BIT #11)

                                ;CONTROL STATUS REGISTER #2  (RMCS2)

000040                          CLR     = BIT5                  ;CONTROLLER CLEAR

                                ;DRIVE STATUS REGISTER (RMDS) (#01)

000001                          OM      = BIT00                 ;OFFSET MODE
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                          LBT     = 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                 ;COMPOSITE ERROR (BIT #14)
100000                          ATA     = 100000                ;ATTENTION ACTIVE (BIT #15)

                                ;ERROR REGISTER #01 (RMER1) (#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 (RMMR1)(#03)

000001                          DMD     = 1                     ;DIAGINOSTIC MODE (BIT #0)
001000                          MUR     = BIT09                 ;MAINTENANCE UNIT READY
040000                          RQB     = BIT14                 ;PORT B REQUEST FLOP
100000                          RQA     = BIT15                 ;PORT A REQUEST FLOP

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

000001                          AT0     = 1                     ;DEVICE 0 (BIT #0)
```

```
000002          AT1     = 2             ;DEVICE 1 (BIT #1)
000004          AT2     = 4             ;DEVICE 2 (BIT #2)
000010          AT3     = 10            ;DEVICE 3 (BIT #3)
000020          AT4     = 20            ;DEVICE 4 (BIT #4)
000040          AT5     = 40            ;DEVICE 5 (BIT #5)
000100          AT6     = 100           ;DEVICE 6 (BIT #6)
000200          AT7     = 200           ;DEVICE 7 (BIT #7)

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

                ;DRIVE TYPE REGISTER (RMDT) (#06)

000001          DT00    = 1             ;DRIVE TYPE NUMBER BIT 1
000002          DT01    = 2             ;DRIVE TYPE NUMBER BIT 2
000004          DT02    = 4             ;DRIVE TYPE NUMBER BIT 3
000010          DT03    = 10            ;DRIVE TYPE NUMBER BIT 4
000020          DT04    = 20            ;DRIVE TYPE NUMBER BIT 5
000040          DT05    = 40            ;DRIVE TYPE NUMBER BIT 6
000100          DT06    = 100           ;DRIVE TYPE NUMBER BIT 7
000200          DT07    = 200           ;DRIVE TYPE NUMBER BIT 8
000400          DT08    = 400           ;DRIVE TYPE NUMBER BIT 9
004000          DRQ     = 4000          ;DRIVE REQUEST REQUIRED (BIT #11)
020000          MOH     = 20000         ;MOVING HEAD (BIT #13)
040000          TAP     = 40000         ;TAPE DRIVE (BIT #14)
100000          NBA     = 100000        ;NOT BLOCK ADDRESSED (BIT #15)

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

000100          SC0     = 100           ;SECTOR COUNT FIELD 0 (BIT #6)
000200          SC1     = 200           ;SECTOR COUNT FIELD 1 (BIT #7)
000400          SC2     = 400           ;SECTOR COUNT FIELD 2 (BIT #8)
001000          SC3     = 1000          ;SECTOR COUNT FIELD 3 (BIT #9)
002000          SC4     = 2000          ;SECTOR COUNT FIELD 4 (BIT #10)

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

000010          DPE     = 10            ;DATA PARITY ERROR  (BIT #3)
000200          DVC     = 200           ;DEVICE CHECK  (BIT #7)
002000          LBC     = 2000          ;LOSS OF BIT CLOCK (BIT #10)
004000          LSC     = 4000          ;LOSS OF SYSTEM CLOCK (BIT #11)
010000          IVC     = 10000         ;INVALID COMMAND (BIT #12)
020000          OPE     = 20000         ;OPERATOR ERROR  (BIT #13)
100000          SKI     = 100000        ;SEEK INCOMPLETE (BIT #14)

                ;OFFSET REGISTER (RMOF) (#11)

000200          OFD     = 200           ;OFFSET FORWARD (BIT #5)
002000          HCI     = 2000          ;HEADER COMPARE INHIBIT (BIT #10)
004000          ECI     = 4000          ;ERROR CORRECTION CODE INHIBIT (BIT #11)
010000          FMT16   = 10000         ;FORMAT BIT (BIT #12

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

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

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

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

                                .SBTTL  DEFINITIONS OF THE RH/RM ADDRESS INDEXES

000000                          RMCS1   = 0             ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
000002                          RMWC    = 2             ;WORD COUNT REGISTER (NOT A DRIVE REG)
000004                          RMBA    = 4             ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
000006                          RMDA    = 6             ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
000010                          RMCS2   = 10            ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
000012                          RMDS    = 12            ;DRIVE STATUS REGISTER (DRIVE REG 01)
000014                          RMER1   = 14            ;ERROR REGISTER #1 (DRIVE REG. 02)
000016                          RMAS    = 16            ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
000020                          RMLA    = 20            ;LOOK AHEAD REGISTER (DRIVE REG. 07)
000022                          RMDB    = 22            ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
000024                          RMMR1   = 24            ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
000026                          RMDT    = 26            ;DRIVE TYPE REGISTER (DRIVE REG. 06)
000030                          RMSN    = 30            ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
000032                          RMOF    = 32            ;OFFSET REGISTER (DRIVE REG. 11)
000034                          RMDC    = 34            ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
000040                          RMMR2   = 40            ;MAINTENANCE REGISTER #2 (DRIVE REG. 14)
000042                          RMER2   = 42            ;ERROR REGISTER #2 (DRIVE REG. 15)
000044                          RMEC1   = 44            ;ECC POSITION REGISTER (DRIVE REG. 16)
000046                          RMEC2   = 46            ;ECC PATTERN REGISTER (DRIVE REG.  17)
```

```
                                        .SBTTL  TRAP CATCHER

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

                                        .SBTTL  STARTING ADDRESS(ES)

  000200  000137  002240                        JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM

  000204  000137  002250                        JMP     @#START1        ;START AND CHANGE THE RH/RM ADDRESS

                                        .SBTTL  ACT11 HOOKS

                                        ;;*****************************************************************
                                        ;HOOKS REQUIRED BY ACT11
          000210                                $SVPC=.                 ;SAVE PC
          000046                                .=46
  000046  066040                                $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
          000052                                .=52
  000052  020000                                .WORD   20000           ;;2)SET LOC.52 TO 20000
          000210                                .=$SVPC                 ;; RESTORE PC
```

```
      0                                 .SBTTL  COMMON TAGS

                                        ;;*****************************************************************
                                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                        ;*USED IN THE 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 ADDRESS
        001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
        001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
        001114     000                  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
        001115     001                  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
        001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
        001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
        001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
        001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
        001132  000000                          .WORD   0
        001134     000                  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
        001135     000                  $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
        001136  000000                          .WORD   0
        001140  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
        001142  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
        001144  177560                  $TKS:   177560                  ;;TTY KBD STATUS
        001146  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
        001150  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
        001152  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
        001154     000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
        001155     002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001156     012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001157     000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001160  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH  ($REG0) WAS OBTAINED
        001162  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
        001164  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
        001166  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
        001170  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
        001172  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
        001174  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
        001176  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
        001200  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
        001202     207     377     377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
        001206     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
        001207     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
        001210     012     000          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                        ;;*****************************************************************
```

```
                                .SBTTL  USER DEFINED TAGS

001212  172540                  $LKCSR: .WORD   172540          ;ADDR OF KW11-P STATUS REGISTER
001214  172542                  $LKCSB: .WORD   172542          ;ADDR OF KW11-P COUNTER BUFFER
001216  000104                  $LPVEC: .WORD   104             ;ADDR OF KW11-P VECTOR
001220  177546                  $LKS:   .WORD   177546          ;ADDR OF KW11-L STATUS REGISTER
001222  000100                  $LLVEC: .WORD   100             ;ADDR OF KW11-L VECTOR
001224  000000                  PORTA:  .WORD   0               ;ADDRESS OF PORT A
001226  000000                  PORTB:  .WORD   0               ;ADDRESS OF PORT B
001230  000000                  PORTC:  .WORD   0               ;ADDRESS OF DIFFERENT DRIVE
001232  000000                  RQSTA:  .WORD   0               ;REQUEST BIT FOR PORT A
001234  000000                  RQSTB:  .WORD   0               ;REQUEST BIT FOR PORT B
001236  000000                  ASR1:   .WORD   0               ;ATA-A OR ATA-B = 1
001240  000000                  PTNBR:  .WORD   0               ;CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
001242  000000                  SEIZPT: .WORD   0               ;CONTAINS THE ADDRESS OF THE SEIZING PORT
001244  000000                  OPPRT:  .WORD   0               ;CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
001246  000000                  TSTNUM: .WORD   0               ;NUMBER OF THE CURRENT TEST
001250  000000                  CKERR:  .WORD   0               ;IF -1, A REGISTER MISCOMPARISON OCCURRED
001252  000000                  NOSEIZ: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
001254  000000                  RELERR: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
001256  000000                  TIME:   .WORD   0               ;ELAPSED TIME COUNTER
001260  000000                  WATCH:  .WORD   0               ;WATCH DOG TIMER LOCATION
001262  000000                  TIMEA:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
001264  000000                  TIMEAP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
001266  000000                  TIMEAM: .WORD   0               ;PORT A TIMEOUT VALUE - 25%
001270  000000                  TIMEB:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
001272  000000                  TIMEBP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
001274  000000                  TIMEBM: .WORD   0               ;PORT B TIME VALUE - 25%
001276  000000                  TIMES:  .WORD   0               ;STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
001300  000000                  KYBCTL: .WORD   0               ;SINGLE TEST INDICATOR
001302  000000                  CHGADR: .WORD   0               ;CHANGE THE RH/RM ADDRESS INDICATOR

                                .SBTTL  RH/RM UNIBUS AND VECTOR ADDRESSES

001304  176700                  $RMADR: .WORD   176700          ;RH/RM UNIBUS ADDRESS
001306  000254                  $RMVEC: .WORD   254             ;INTERRUPT VECTOR ADDRESS
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


  001310                        $ERRTB:
                                ;ERROR 1

  001310  072654                        EM1     ;WRONG DRIVE TYPE
  001312  077306                        DH1
  001314  101176                        DT1
  001316  101464                        DF1

                                ;ERROR 2

  001320  072675                        EM2     ;DRIVE NOT ON LINE
  001322  077306                        DH1
  001324  101176                        DT1
  001326  101464                        DF1

                                ;ERROR 3

  001330  072717                        EM3     ;SERIAL NUMBERS NOT THE SAME
  001332  077357                        DH3
  001334  101212                        DT3
  001336  101464                        DF1

                                ;ERROR 4

  001340  073001                        EM4     ;DRIVE NOT SEIZED BY PORT 'N'
  001342  077426                        DH4
  001344  101260                        DT7
  001346  101477                        DF7

                                ;ERROR 5

  001350  073032                        EM5     ;WRONG STATUS SEEN BY THE SEIZING PORT
  001352  077551                        DH5
  001354  101226                        DT5
  001356  101471                        DF5

                                ;ERROR 6

  001360  073100                        EM6     ;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED
  001362  100021                        DH13
  001364  101300                        DT13
  001366  101471                        DF5
```

```
                                        ;ERROR 7

001370  073200                                  EM7     ;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
001372  077625                                  DH7
001374  101260                                  DT7
001376  101477                                  DF7

                                        ;ERROR 10

001400  073261                                  EM10    ;REGISTER CONTENTS INCORRECT
001402  077551                                  DH5
001404  101226                                  DT5
001406  101471                                  DF5

                                        ;ERROR 11

001410  073311                                  EM11    ;CONTROL BUS PARITY ERROR WHILE READING REGISTER
001412  077750                                  DH11
001414  101176                                  DT1
001416  101464                                  DF1

                                        ;ERROR 12

001420  073375                                  EM12    ;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
001422  100515                                  DH36
001424  101366                                  DT37
001426  101512                                  DF36

                                        ;ERROR 13

001430  073445                                  EM13    ;'VOLUME VALID' BIT NOT SET BY READIN PRESET
001432  100021                                  DH13
001434  101300                                  DT13
001436  101471                                  DF5

                                        ;ERROR 14

001440  073532                                  EM14    ;'VOLUME VALID' SET ON THE OPPOSITE PORT
001442  100021                                  DH13
001444  101300                                  DT13
001446  101471                                  DF5

                                        ;ERROR 15

001450  073575                                  EM15    ;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
001452  077625                                  DH7
001454  101260                                  DT7
001456  101477                                  DF7

                                        ;ERROR 16

001460  073654                                  EM16    ;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
001462  077625                                  DH7
001464  101260                                  DT7
001466  101477                                  DF7
```

```
                                        ;ERROR 17
001470  073727                                  EM17    ;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
001472  077625                                  DH7
001474  101260                                  DT7
001476  101477                                  DF7

                                        ;ERROR 20

001500  074006                                  EM20    ;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
001502  100515                                  DH36
001504  101366                                  DT37
001506  101512                                  DF36

                                        ;ERROR 21

001510  074066                                  EM21    ;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
001512  100515                                  DH36
001514  101366                                  DT37
001516  101512                                  DF36

                                        ;ERROR 22

001520  074141                                  EM22    ;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
001522  100141                                  DH22
001524  101316                                  DT22
001526  101506                                  DF31

                                        ;ERROR 23

001530  074226                                  EM23    ;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
001532  100237                                  DH23
001534  101330                                  DT23
001536  101464                                  DF1

                                        ;ERROR 24

001540  074274                                  EM24    ;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
001542  100237                                  DH23
001544  101330                                  DT23
001546  101464                                  DF1


                                        ;ERROR 25

001550  074353                                  EM25    ;TIMEOUT ONE-SHOT DID NOT RETRIGGER
001552  100515                                  DH36
001554  101356                                  DT36
001556  101512                                  DF36


                                        ;ERROR 26

001560  074416                                  EM26    ;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
001562  100141                                  DH22
001564  101316                                  DT22
001566  101506                                  DF31
```

```
                                ;ERROR 27
001570  074503                          EM27    ;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
001572  077625                          DH7
001574  101260                          DT7
001576  101477                          DF7

                                ;ERROR 30

001600  074561                          EM30    ;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
001602  100515                          DH36
001604  101356                          DT36
001606  101512                          DF36

                                ;ERROR 31

001610  074656                          EM31    ;DRIVE NOT SEIZED BY PORT AFTER RELEASE WIITH REQUEST SET
001612  100416                          DH31
001614  101344                          DT31
001616  101506                          DF31

                                ;ERROR 32

001620  074733                          EM32    ;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
001622  077551                          DH5
001624  101226                          DT5
001626  101471                          DF5

                                ;ERROR 33

001630  075004                          EM33    ;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED
001632  100515                          DH36
001634  101356                          DT36
001636  101512                          DF36

                                ;ERROR 34

001640  075106                          EM34    ;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED
001642  100515                          DH36
001644  101356                          DT36
001646  101512                          DF36

                                ;ERROR 35

001650  075211                          EM35    ;DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT ONE SHOT
001652  100515                          DH36
001654  101366                          DT37
001656  101512                          DF36

                                ;ERROR 36

001660  075270                          EM36    ;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
001662  100515                          DH36
001664  101356                          DT36
001666  101512                          DF36
```

```
                                ;ERROR 37

                                        EM37    ;DRIVE IS NON-EXISTENT
001670  075342                          DH36
001672  100515                          DT37
001674  101366                          DF36
001676  101512

                                ;ERROR 40

                                        EM40    ;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
001700  075410                          DH1
001702  077306                          DT23
001704  101330                          DF1
001706  101464

                                ;ERROR 41

                                        EM41    ;TIMEOUT CLEARED ATTENTION BIT
001710  075465                          DH23
001712  100237                          DT23
001714  101330                          DF1
001716  101464

                                ;ERROR 42

                                        EM42    ;DRIVE NOT IN NEUTRAL OR SEIZED
001720  075527                          DH42
001722  100544                          DT42
001724  101376                          DF42
001726  101515

                                ;ERROR 43

                                        EM43    ;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
001730  075615                          DH42
001732  100544                          DT42
001734  101376                          DF42
001736  101515

                                ;ERROR 44

                                        EM44    ;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
001740  075672                          DH44
001742  100563                          DT31
001744  101344                          DF31
001746  101506

                                ;ERROR 45

                                        EM45    ;PORT SELECT SWITCH ON DRIVE NOT IN 'A/B'
001750  075747                          DH1
001752  077306                          DT1
001754  101176                          DF1
001756  101464

                                ;ERROR 46

                                        EM46    ;CAN'T ACCESS DRIVE THROUGH EITHER PORT
001760  076021                          DH46
001762  100661                          DT46
001764  101404                          DF31
001766  101506

                                ;ERROR 47
```

```
271
272 001770  076070                          EM47    ;ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
273 001772  100237                          DH23
274 001774  101330                          DT23
275 001776  101464                          DF1
276
277                                 ;ERROR 50
278
279 002000  076156                          EM50    ;ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMAND
280 002002  100021                          DH13
281 002004  101300                          DT13
282 002006  101471                          DF5
283
284                                 ;ERROR 51
285
286 002010  076240                          EM51    ;ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
287 002012  077551                          DH5
288 002014  101226                          DT5
289 002016  101471                          DF5
290
291                                 ;ERROR 52
292
293 002020  076327                          EM52    ;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
294 002022  100021                          DH13
295 002024  101300                          DT13
296 002026  101471                          DF5
297
298                                 ;ERROR 53
299
300 002030  076422                          EM53    ;CAN'T READ ATTN BIT FROM OPPOSITE PORT
301 002032  100237                          DH23
302 002034  101176                          DT1
303 002036  101464                          DF1
304
305                                 ;ERROR 54
306
307 002040  076503                          EM54    ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
308 002042  100141                          DH22
309 002044  101416                          DT54
310 002046  101506                          DF31
311
312                                 ;ERROR 55
313
314 002050  076576                          EM55    ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
315 002052  100756                          DH55
316 002054  101430                          DT55
317 002056  101517                          DF55
318
319                                 ;ERROR 56
320
321 002060  076643                          EM56    ;RH/RM DIDN'T RESPOND TO ADDRESSING
322 002062  101034                          DH56
323 002064  101442                          DT56
324 002066  101523                          DF56
325
326
327                                 ;ERROR  57
```

```
002070  076706                          EM57    ;PORT REQUEST FLOPS WRONG
002072  101043                          DH57
002074  101446                          DT57
002076  101471                          DF5

                                ;ERROR  60

002100  076747                          EM60    ;ATTENTION BITS NOT RESET BY RMAS
002102  077551                          DH5
002104  101226                          DT5
002106  101471                          DF5

                                ;ERROR  61

002110  077013                          EM61    ;ATTENTION NOT RESET BY GO
002112  100237                          DH23
002114  101330                          DT23
002116  101464                          DF1

                                ;ERROR  62

002120  077045                          EM62    ;ATTENTION RESET BY GO WHEN NOT SEIZED
002122  100021                          DH13
002124  101300                          DT13
002126  101471                          DF5

                                ;ERROR  63

002130  077113                          EM63    ;DRIVE SEIZED BY 'MUR' CHANGE IN RMMR1
002132  100515                          DH36
002134  101356                          DT36
002136  101512                          DF36

                                ;ERROR  64

002140  077161                          EM64    ;ATTENTION NOT SET BY 'MUR' CHANGE IN RMMR1
002142  077625                          DH7
002144  101260                          DT7
002146  101477                          DF7

                                ;ERROR  65

002150  077234                          EM65    ;VV NOT RESET BY RMMR1 'MUR' IN RMMR1
002152  077551                          DH5
002154  101226                          DT5
002156  101471                          DF5
```

```
                                        ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS

002160  011600                          BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
002162  005740                                  TST     -(R0)           ;ADJUST PC -2
002164  022626                                  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
002166  104401  002174                          TYPE    ,65$            ;;TYPE ASCIZ STRING
002172  000417                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
002232                                  64$:
002232  010046                                  MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
002234  104402                                  TYPOC
002236  000240                                  NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                        ;TO STOP ON UNEXPECTED TIMEOUT.


                                        .SBTTL  START OF PROGRAM

002240  000240                          START:  NOP
002242  005037  001302                          CLR     CHGADR          ;CLEAR THE 'CHANGE RH/RM ADDRESS' INDICATOR
002246  000403                                  BR      START2          ;GO TO THE START
002250  012737  177777  001302          START1: MOV     #-1,CHGADR      ;SET THE 'CHANGE RH/RM ADDRESS' INDICATOR

002256  005227  000000                  START2: INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
002262  001375                                  BNE     .-4             ;OF WORD
002264  000005                                  RESET                   ;CLEAR THE WORLD

                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002266  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002272  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002274  022706  001140                          CMP     #SWR,R6         ;;DONE?
002300  001374                                  BNE     .-6             ;;LOOP BACK IF NO
002302  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
002306  012737  066314  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002314  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002322  012737  066676  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002330  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002336  012737  072234  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002344  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
002352  013737  065676  065670                  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002360  005037  001176                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002364  005037  001200                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002370  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002376  012737  002376  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002404  012737  002404  001110                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002412  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002416  012737  002452  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002424  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002432  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002440  022777  177777  176472                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002446  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002450  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
002452  012716  002460                  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002456  000002                                  RTI
```

INITIALIZE THE COMMON TAGS

```
        002460  012737  000176  001140  65$:    MOV     #SWREG,SWR        ;;POINT TO SOFTWARE SWR
        002466  012737  000174  001142           MOV     #DISPREG,DISPLAY
        002474  012637  000004          66$:     MOV     (SP)+,@#ERRVEC    ;;RESTORE ERROR VECTOR

   24                                   ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
   25 002500  012737  002160  000004           MOV     #BADTMO,ERRVEC    ;SETUP FOR UNEXPECTED TIMEOUT
   26 002506  012737  000300  000006           MOV     #PR6,ERRVEC+2     ;LEVEL 6
   27
   28                                   .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
        002514  005227  177777                   INC     #-1               ;;FIRST TIME?
        002520  001032                           BNE     67$               ;;BRANCH IF NO
        002522  022737  066040  000042           CMP     #$ENDAD,@#42      ;;ACT-11?
        002530  001426                           BEQ     67$               ;;BRANCH IF YES
        002532  104401  002540                   TYPE    ,68$              ;;TYPE ASCIZ STRING
        002536  000423                           BR      67$               ;;GET OVER THE ASCIZ
                                        ;;68$:   .ASCIZ  <CRLF>@CZRNHA0 - RM80 DUAL PORT TEST, PT 1@<CRLF>
        002606                          67$:
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
        002606  005737  000042                   TST     @#42              ;;ARE WE RUNNING UNDER XXDP/ACT?
        002612  001006                           BNE     69$               ;;BRANCH IF YES
        002614  023727  001140  000176           CMP     SWR,#SWREG        ;;SOFTWARE SWITCH REG SELECTED?
        002622  001005                           BNE     70$               ;;BRANCH IF NO
        002624  104406                           GTSWR                     ;;GET SOFT-SWR SETTINGS
        002626  000403                           BR      70$
        002630  112737  000001  001134  69$:     MOVB    #1,$AUTOB         ;;SET AUTO-MODE INDICATOR
        002636                          70$:
   29 002636  004737  070540                     JSR     PC,$TKINT         ;SETUP THE TTY KEYBOARD
   30 002642  004737  003236           1$:       JSR     PC,CHANGE         ;CHECK/CHANGE THE RH/RM ADDRESS
   31 002646  104401  072322                     TYPE    ,ENTERA           ;ENTER DRIVE ADDRESS
   32 002652  104412                             RDOCT                     ;GET THE ADDRESS
   33 002654  012637  001224                     MOV     (SP)+,PORTA       ;STORE THE ADDRESS
   34 002660  023727  001224  000007             CMP     PORTA,#7          ;SEE IF ADDRESS TOO LARGE
   35 002666  101403                             BLOS    2$                ;BR IF NOT
   36 002670  104401  072351                     TYPE    ,ADRERR           ;TYPE ADDRESS ERROR MESSAGE
   37 002674  000762                             BR      1$                ;TRY AGAIN
   38 002676  013737  001224  001226   2$:       MOV     PORTA,PORTB       ;GENERATE THE PORT B ADDRESS
   39 002704  005237  001226                     INC     PORTB             ;INCREMENT THE ADDRESS
   40 002710  042737  000016  001226             BIC     #16,PORTB         ;LEAVE BIT 0
   41 002716  013746  001224                     MOV     PORTA,-(SP)       ;PUT PORT A ADDRESS ON THE STACK
   42 002722  042716  177771                     BIC     #^C6,(SP)         ;SAVE BITS 1 & 2
   43 002726  052637  001226                     BIS     (SP)+,PORTB       ;SET BITS 1 & 2 IN PORT B ADDRESS
   44 002732  104401  072374                     TYPE    ,PORTAIS          ;'PORT A ADDRESS IS '
   45 002736  013746  001224                     MOV     PORTA,-(SP)       ;;SAVE PORTA FOR TYPEOUT
                                                                           ;;TYPE PORT A ADDRESS
        002742  104403                           TYPOS                     ;;GO TYPE--OCTAL ASCII
        002744     001                           .BYTE   1                 ;;TYPE 1 DIGIT(S)
        002745     000                           .BYTE   0                 ;;SUPPRESS LEADING ZEROS
   46 002746  104401  072423                     TYPE    ,PORTBIS          ;'PORT B ADDRESS IS '
   47 002752  013746  001226                     MOV     PORTB,-(SP)       ;;SAVE PORTB FOR TYPEOUT
                                                                           ;;TYPE PORT B ADDRESS
        002756  104403                           TYPOS                     ;;GO TYPE--OCTAL ASCII
        002760     001                           .BYTE   1                 ;;TYPE 1 DIGIT(S)
        002761     000                           .BYTE   0                 ;;SUPPRESS LEADING ZEROS
   48 002762  104401  001207                     TYPE    ,$CRLF            ;ANOTHER CR-LF
   49 002766  013737  001224  001230             MOV     PORTA,PORTC       ;GENERATE ADDRESS OF DRIVE NOT TESTED
   50 002774  062737  000006  001230             ADD     #6,PORTC          ;COMPLEMENT SOME BITS
```

```
003002  042737  177770  001230          BIC     #^C7,PORTC      ;SAVE ONLY LOWER BITS
003010  013701  001224                  MOV     PORTA,R1        ;USE PORT A ADDRESS AS INDEX
003014  116137  101640  001236          MOVB    ATABIT(R1),ASR1  ;GET ATTENTION BIT FOR DRIVE
003022  005037  001262                  CLR     TIMEA           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
003026  005037  001264                  CLR     TIMEAP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
003032  005037  001270                  CLR     TIMEB           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
003036  005037  001272                  CLR     TIMEBP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
003042  004737  066060                  JSR     PC,CKCLK        ;SETUP CLOCK
003046  000137  003062                  JMP     EXEC            ;CLOCK HAS BEEN STARTED
003052  104401  072452                  TYPE    ,NOCLOCK        ;NO CLOCK ON SYSTEM
003056  000000                  3$:     HALT                    ;FATAL ERROR
003060  000776                          BR      3$              ;INTERLOCK THE HALT

                                ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR

003062  000005                  EXEC:   RESET                   ;CLEAR EVERYTHING
003064  005037  177776                  CLR     PS              ;CLEAR THE PROCESSOR STATUS WORD
003070  104401  001207                  TYPE    ,$CRLF          ;CR-LF
003074  013700  001304                  MOV     $RMADR,R0       ;RH/RM ADDRESS FOR INDEXING
003100  012706  001100                  MOV     #STACK,SP       ;LOAD STACK POINTER
003104  004737  066060                  JSR     PC,CKCLK        ;START THE CLOCK
003110  000240                          NOP                     ;RETURN IF NO CLOCK
003112  004737  070540                  JSR     PC,$TKINT       ;INITIALIZE THE KEYBOARD
003116  005037  001300                  CLR     KYBCTL          ;CLEAR SINGLE TEST INDICATOR
003122  005037  001100                  CLR     $PASS           ;CLEAR THE PASS COUNT
003126  112737  000001  001115          MOVB    #1,$ERMAX       ;SET ERROR MAX TO 1
003134  012737  003134  001106          MOV     #.,$LPADR       ;INITIAL SETTING FOR LOOP ADDRESS
003142  012737  003142  001110          MOV     #.,$LPERR       ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
003150  104401  072510          1$:     TYPE    ,TESTNO         ;ASK FOR TEST NUMBER
003154  104412                          RDOCT                   ;GET THE NUMBER
003156  012601                          MOV     (SP)+,R1        ;PUT ENTRY INTO R1
003160  001002                          BNE     2$              ;BR IF NOT ZERO
003162  000137  003346                  JMP     TST1AA          ;ENTER ZERO - PERFORM ALL TESTS

003166  020137  101650          2$:     CMP     R1,MAXTN        ;SEE IF NUMBER GREATER THAN MAXIMUM
003172  003403                          BLE     3$              ;BR IF LESS OR EQUAL
003174  104401  072530                  TYPE    ,BADNO          ;BAD ENTRY
003200  000763                          BR      1$              ;TRY AGAIN
003202  005301                  3$:     DEC     R1              ;DECREMENT ENTRY
003204  006301                          ASL     R1              ;SHIFT IT LEFT
003206  005237  001300                  INC     KYBCTL          ;SET SINGLE TEST INDICATOR
003212  012737  000001  001104          MOV     #1,$ICNT        ;PRESET ITERATION COUNT
003220  012746  000240                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
003224  012746  003232                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003230  000002                          RTI                     ;;POP NEW PC AND PS
003232                          64$:
003232  000171  101524                  JMP     @TSTADR(R1)     ;JUMP TO TEST

                                ;CHANGE THE RH/RM UNIBUS ADDRESS USED BY THE PROGRAM

003236  005737  001302          CHANGE: TST     CHGADR          ;CHANGE THE ADDRESS ?
003242  001421                          BEQ     3$              ;BR IF NOT
003244  005037  001302                  CLR     CHGADR          ;CLEAR THE INDICATOR
003250  104401  072570          1$:     TYPE    ,ADDRIS         ;TYPE OUT WHAT THE PRESENT ADDRESS IS
003254  013746  001304                  MOV     $RMADR,-(SP)    ;PUT THE ADDRESS ON THE STACK
003260  104402                          TYPOC                   ;TYPE THE ACTUAL ADDRESS
003262  104401  001207                  TYPE    ,$CRLF          ;CR-LF
```

```
003266  104401  072625                  TYPE    ,NTRH           ;ASK FOR NEW ADDRESS
003272  104412                          RDOCT
003274  005716                          TST     (SP)            ;0 OR 'CR' ENTERED ?
003276  001402                          BEQ     2$              ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
003300  011637  001304                  MOV     (SP),$RMADR     ;NEW RH/RM ADDRESS
003304  005726                  2$:     TST     (SP)+           ;CORRECT THE STACK POINTER
003306  012737  003326  000004  3$:     MOV     #4$,@#4         ;LOAD TRAP ADDRESS
003314  013700  001304                  MOV     $RMADR,R0       ;GET RH/RM ADDRESS
003320  005760  000002                  TST     RMWC(R0)        ;RESPONDS AT THAT ADDRESS ?
003324  000404                          BR      5$              ;BR IF YES
003326                          4$:
003326  104056                          EMT     56
003330  062706  000004                  ADD     #4,SP           ;RESET THE STACK POINTER
003334  000745                          BR      1$              ;GET ADDRESS AGAIN
003336  012737  000006  000004  5$:     MOV     #6,@#4          ;RESTORE THE VECTOR
003344  000207                          RTS     PC              ;RETURN

                                TST1AA: MOV     $RMADR,R0       ;;RESTORE R0 AFTER END OF PASS
003346  013700  001304                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
003352  012746  000240                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003356  012746  003364                  RTI                     ;;POP NEW PC AND PS
003362  000002                  64$:
003364

                                ;;*****************************************************************
                                ;*TEST 1         NEUTRAL ACCESS TEST
                                ;*
                                ;*VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS
                                ;*
                                ;*  A.  SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
                                ;*      DRIVE IS A DUAL PORT RM80, THAT THE DRIVE IS ONLINE (RMDS 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.
                                ;*
                                ;*****************************************************************
003364                          TST1:
003364  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
003370  001406                          BEQ     2$              ;BR IF NOT
003372  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
003374  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
003400  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
003406  012737  003422  001106  2$:     MOV     #TEST1,$LPADR   ;SETUP SCOPE LOOP ADDRESS
003414  012737  003422  001110          MOV     #TEST1,$LPERR   ;SETUP ERROR LOOP ADDRESS
003422                          TEST1:
003422  112737  000001  001102          MOVB    #1,$TSTNM       ;MOVE #1 TO TEST NUMBER
003430  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
003434  012737  000001  001176          MOV     #1,$TIMES       ;;DO 1 ITERATION

003442  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;INITIALIZE THE MASSBUS

                                ;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

003450  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
003456  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003464  005760  000012                  TST     RMDS(R0)        ;SEE IF DRIVE (PORT A) PRESENT
```

```
003470  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
003474  016037  000010  001126           MOV     RMCS2(R0),$BDDAT  ;GET CONTENTS OF RMCS2
003502  012737  000010  001122           MOV     #RMCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003510  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
003514  005037  001124                   CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
003520  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
003526  042737  167777  001164           BIC     #^CNED,$TMP0  ;SAVE SPECIFIED BITS
003534  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
003542  001414                           BEQ     64$             ;BR IF OK
003544  013737  001126  001174           MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
003552  042737  010000  001174           BIC     #NED,$TMP4      ;CLEAR THE MASKED BITS
003560  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
003566  104037                           EMT     37
003570  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
003574  000240                  64$:     NOP
003576  005737  001250                   TST     CKERR           ;WAS 'NED' SET ?
003602  001403                           BEQ     .+10            ;BR IF NOT
003604  012760  000040  000010           MOV     #CLR,RMCS2(R0)  ;ISSUE MASSBUS INIT TO CLEAR 'NED'
003612  113760  001226  000010           MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
003620  013737  001226  001240           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003626  005760  000012                   TST     RMDS(R0)        ;SEE IF DRIVE (PORT B) PRESENT
003632  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
003636  016037  000010  001126           MOV     RMCS2(R0),$BDDAT  ;GET CONTENTS OF RMCS2
003644  012737  000010  001122           MOV     #RMCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003652  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
003656  005037  001124                   CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
003662  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
003670  042737  167777  001164           BIC     #^CNED,$TMP0  ;SAVE SPECIFIED BITS
003676  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
003704  001414                           BEQ     66$             ;BR IF OK
003706  013737  001126  001174           MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
003714  042737  010000  001174           BIC     #NED,$TMP4      ;CLEAR THE MASKED BITS
003722  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
003730  104037                           EMT     37
003732  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
003736  000240                  66$:     NOP
003740  005737  001250                   TST     CKERR           ;WAS 'NED' SET ?
003744  001403                           BEQ     .+10            ;BR IF NOT
003746  012760  000040  000010           MOV     #CLR,RMCS2(R0)  ;ISSUE MASSBUS INIT TO CLEAR 'NED'

                                ;CONFIRM THAT DRIVE IS AN RM80 AND IS DUAL PORT

003754  113760  001224  000010           MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
003762  013737  001224  001240           MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003770  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
003774  016037  000026  001126           MOV     RMDT(R0),$BDDAT  ;GET CONTENTS OF RMDT
004002  012737  000026  001122           MOV     #RMDT,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004010  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
004014  012737  024026  001124           MOV     #024026,$GDDAT  ;WHAT REGISTER SHOULD BE
004022  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
004030  001403                           BEQ     68$             ;BR IF OK
004032  104001                           EMT     1
004034  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
004040  000240                  68$:     NOP
004042  113760  001226  000010           MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
004050  013737  001226  001240           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004056  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
        004062  016037  000026  001126          MOV     RMDT(R0),$BDDAT  ;GET CONTENTS OF RMDT
        004070  012737  000026  001122          MOV     #RMDT,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        004076  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        004102  012737  024026  001124          MOV     #024026,$GDDAT  ;WHAT REGISTER SHOULD BE
        004110  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
        004116  001403                          BEQ     70$             ;BR IF OK
        004120  104001                          EMT     1
        004122  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        004126  000240                  70$.    NOP
    157
    158                                 ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
    159
    164 004130  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
        004136  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        004144  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        004150  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
        004156  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        004164  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        004170  012737  001000  001124          MOV     #PGM,$GDDAT  ;WHAT REGISTER SHOULD BE
        004176  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
        004204  042737  176777  001164          BIC     #^CPGM,$TMP0  ;SAVE SPECIFIED BITS
        004212  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
        004220  001414                          BEQ     72$             ;BR IF OK
        004222  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
        004230  042737  001000  001174          BIC     #PGM,$TMP4      ;CLEAR THE MASKED BITS
        004236  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
        004244  104045                          EMT     45
        004246  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        004252  000240                  72$:    NOP
        004254  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        004260  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
        004266  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        004274  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        004300  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
        004306  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
        004314  042737  167177  001164          BIC     #^C10600,$TMP0  ;SAVE SPECIFIED BITS
        004322  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
        004330  001414                          BEQ     74$             ;BR IF OK
        004332  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
        004340  042737  010600  001174          BIC     #10600,$TMP4    ;CLEAR THE MASKED BITS
        004346  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
        004354  104002                          EMT     2
        004356  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        004362  000240                  74$:    NOP
        004364  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
        004372  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        004400  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        004404  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
        004412  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        004420  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        004424  012737  001000  001124          MOV     #PGM,$GDDAT  ;WHAT REGISTER SHOULD BE
        004432  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
        004440  042737  176777  001164          BIC     #^CPGM,$TMP0  ;SAVE SPECIFIED BITS
        004446  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
        004454  001414                          BEQ     76$             ;BR IF OK
        004456  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
        004464  042737  001000  001174          BIC     #PGM,$TMP4      ;CLEAR THE MASKED BITS
```

```
004472  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
004500  104045                          EMT     45
004502  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
004506  000240                  76$:    NOP
004510  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
004514  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
004522  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004530  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
004534  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
004542  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
004550  042737  167177  001164          BIC     #^C10600,$TMP0  ;SAVE SPECIFIED BITS
004556  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
004564  001414                          BEQ     78$             ;BR IF OK
004566  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
004574  042737  010600  001174          BIC     #10600,$TMP4    ;CLEAR THE MASKED BITS
004602  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
004610  104002                          EMT     2
004612  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
004616  000240                  78$:    NOP

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

004620  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
004626  016037  000030  001124          MOV     RMSN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
004634  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
004642  016037  000030  001126          MOV     RMSN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
004650  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ARE THEY THE SAME ?
004656  001406                          BEQ     1$              ;BR IF THEY ARE
004660  104003                          EMT     3
004662  032777  100000  174250          BIT     #SW15,@SWR      ;HALT ON ERROR ?
004670  001001                          BNE     1$              ;BR IF SET - PROGRAM HAS ALREADY HALTED
004672  000000                          HALT                    ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
004674  000004                  1$:     SCOPE                   ;LOOP ?

                                ;;*****************************************************************
                                ;*TEST 2         PORT 'A' SEIZE/TIMEOUT TEST
                                ;*
                                ;*VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT
                                ;*      IT CAN BE RELEASED BY THE ONE SECOND TIMER.
                                ;*
                                ;*  A.  WRITE 0'S INTO RMDA THROUGH PORT 'A'; VERIFY THAT THE DRIVE
                                ;*      HAS BEEN SEIZED.
                                ;*
                                ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B';
                                ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
                                ;*
                                ;*  C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
                                ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
                                ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
                                ;*      NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.
                                ;*
                                ;;*****************************************************************
004676                          TST2:
004676  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
004702  001406                          BEQ     2$              ;BR IF NOT
004704  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
```

```
        004706  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
        004712  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
        004720  012737  004734  001106  2$:     MOV     #TEST2,$LPADR   ;SETUP SCOPE LOOP ADDRESS
        004726  012737  004734  001110          MOV     #TEST2,$LPERR   ;SETUP ERROR LOOP ADDRESS
        004734                          TEST2:
        004734  112737  000002  001102          MOVB    #2,$TSTNM       ;MOVE #2 TO TEST NUMBER
        004742  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
        004746  012737  000002  001176          MOV     #2.,$TIMES      ;;DO 2. ITERATIONS
    199
    267 004754  012737  000240  177776          MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
        004762  005037  001262                  CLR     TIMEA           ;CLEAR TIMEOUT VALUE FOR PORT A
        004766  005037  001264                  CLR     TIMEAP  ;CLEAR UPPER TIMEOUT TOLERANCE
        004772  005037  001266                  CLR     TIMEAM  ;CLEAR LOWER TIMEOUT TOLERANCE

                                        ;START THE TIMER

        004776  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
        005002  012737  003720  001260          MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS

                                        ;SEIZE THE DRIVE THROUGH PORT A

        005010  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
        005016  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
        005024  005060  000006                  CLR     RMDA(R0)        ;WRITE RMDA
        005030  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
        005036  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        005044  013737  001226  001244          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
        005052  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
        005060  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
        005064  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
        005072  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
        005076  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
        005104  001403                          BEQ     64$             ;BR IF IT IS
        005106  104004                          EMT     4
        005110  000137  006242                  JMP     5$              ;BYPASS REST OF THE SUBTEST
        005114                          64$:
        005114  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
        005122  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        005130  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
        005136  042737  ^20001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
        005144  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;EXPECTED STATUS
        005152  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
        005160  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
        005164  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
        005172  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
        005200  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
        005206  001401                          BEQ     65$             ;BR IF THEY ARE
        005210  104005                          EMT     5
        005212  000240                  65$:    NOP

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

        005214  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
        005222  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        005230  016046  000046                  MOV     RMEC2(R0),-(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
        005234  016046  000044                  MOV     RMEC1(R0),-(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
        005240  016046  000030                  MOV     RMSN(R0),-(SP)  ;STORE REGISTER RMSN, PORT B, FOR CHECK
```

```
005244  016046  000034                  MOV     RMDC(R0),-(SP)   ;STORE REGISTER RMDC, PORT B, FOR CHECK
005250  016046  000032                  MOV     RMOF(R0),-(SP)   ;STORE REGISTER RMOF, PORT B, FOR CHECK
005254  016046  000042                  MOV     RMER2(R0),-(SP)  ;STORE REGISTER RMER2, PORT B, FOR CHECK
005260  016046  000020                  MOV     RMLA(R0),-(SP)   ;STORE REGISTER RMLA, PORT B, FOR CHECK
005264  016046  000026                  MOV     RMDT(R0),-(SP)   ;STORE REGISTER RMDT, PORT B, FOR CHECK
005270  016046  000006                  MOV     RMDA(R0),-(SP)   ;STORE REGISTER RMDA, PORT B, FOR CHECK
005274  016046  000024                  MOV     RMMR1(R0),-(SP)  ;STORE REGISTER RMMR1, PORT B, FOR CHECK
005300  016046  000014                  MOV     RMER1(R0),-(SP)  ;STORE REGISTER RMER1, PORT B, FOR CHECK

                                ;WAIT FOR PORT A TO TIMEOUT

005304  005760  000012          1$:     TST     RMDS(R0)         ;WAIT FOR THE DRIVE TO TIMEOUT
005310  001006                          BNE     2$               ;BR WHEN TIMEOUT OCCURS
005312  005737  001260                  TST     WATCH            ;CHECK WATCH
005316  001372                          BNE     1$               ;BR IF NOT ZERO
005320  104036                          EMT     36
005322  000137  005726                  JMP     4$               ;BYPASS TIMEOUT TIME CHECK
005326  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS    ;SET PRIORITY TO 7 TO STOP CLOCK
005334  013737  001256  001262          MOV     TIME,TIMEA       ;SAVE THE ELAPSED TIME FOR PORT A
005342  004537  066254                  JSR     R5,TOLER         ;CALCULATE THE TOLERANCE
005346  001262                          .WORD   TIMEA            ;TIMEOUT VALUE FOR PORT A
005350  012637  001264                  MOV     (SP)+,TIMEAP     ;+25% TOLERANCE
005354  012637  001266                  MOV     (SP)+,TIMEAM     ;-25% TOLERANCE

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

005360  023727  001256  000764          CMP     TIME,#500.       ;WAS MEASURED TIME AT LEAST 500 MS?
005366  103001                          BHIS    3$               ;BR IF IT  WAS
005370  104055                          EMT     55

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

005372  012737  000240  177776  3$:     MOV     #<5*32.>,@#PS    ;RESTORE PRIORITY TO 5

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

005400  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
005404  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
005412  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
005416  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
005424  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
005432  016037  000012  001170          MOV     RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
005440  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
005446  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
005454  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005462  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
005470  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
005476  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
005504  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
005512  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005520  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
005526  001006                          BNE     66$              ;BR IF NOT
005530  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
005534  001037                          BNE     68$              ;BR IF NOT
005536  104046                          EMT     46
005540  000137  005724                  JMP     70$              ;BYPASS THE REST OF THE CHECKS
005544  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
```

```
005552  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005560  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
005566  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
005572  001414                          BEQ     67$              ;BR IF ZERO
005574  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005602  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
005610  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
005616  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
005622  001004                          BNE     68$              ;BR IF NOT
005624  012737  177777  001254  67$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
005632  104022                          EMT     22
005634  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
005642  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
005650  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
005656  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
005664  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
005666  104007                          EMT     7
005670  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
005676  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
005704  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
005712  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
005720  001401                          BEQ     70$              ;BR IF OK
005722  104007                          EMT     7
005724  000240                  70$:    NOP

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

005726  013737  001226  001240  4$:     MOV     PORTB,PTNBR      ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
005734  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMER1
005740  062737  000014  001122          ADD     #RMER1,$BDADR    ;ADDRESS OF RMER1 FOR TYPEOUT
005746  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMER1
005752  001401                          BEQ     .+4              ;CONTENTS ZERO ?
005754  104006                          EMT     6
005756  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMMR1
005762  062737  000024  001122          ADD     #RMMR1,$BDADR    ;ADDRESS OF RMMR1 FOR TYPEOUT
005770  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMMR1
005774  001401                          BEQ     .+4              ;CONTENTS ZERO ?
005776  104006                          EMT     6
006000  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMDA
006004  062737  000006  001122          ADD     #RMDA,$BDADR     ;ADDRESS OF RMDA FOR TYPEOUT
006012  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMDA
006016  001401                          BEQ     .+4              ;CONTENTS ZERO ?
006020  104006                          EMT     6
006022  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMDT
006026  062737  000026  001122          ADD     #RMDT,$BDADR     ;ADDRESS OF RMDT FOR TYPEOUT
006034  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMDT
006040  001401                          BEQ     .+4              ;CONTENTS ZERO ?
006042  104006                          EMT     6
006044  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMLA
006050  062737  000020  001122          ADD     #RMLA,$BDADR     ;ADDRESS OF RMLA FOR TYPEOUT
006056  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMLA
006062  001401                          BEQ     .+4              ;CONTENTS ZERO ?
006064  104006                          EMT     6
006066  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMER2
006072  062737  000042  001122          ADD     #RMER2,$BDADR    ;ADDRESS OF RMER2 FOR TYPEOUT
006100  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMER2
```

```
006104  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006106  104006                          EMT     6
006110  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMOF
006114  062737  000032  001122          ADD     #RMOF,$BDADR    ;ADDRESS OF RMOF FOR TYPEOUT
006122  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMOF
006126  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006130  104006                          EMT     6
006132  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMDC
006136  062737  000034  001122          ADD     #RMDC,$BDADR    ;ADDRESS OF RMDC FOR TYPEOUT
006144  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMDC
006150  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006152  104006                          EMT     6
006154  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMSN
006160  062737  000030  001122          ADD     #RMSN,$BDADR    ;ADDRESS OF RMSN FOR TYPEOUT
006166  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMSN
006172  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006174  104006                          EMT     6
006176  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMEC1
006202  062737  000044  001122          ADD     #RMEC1,$BDADR   ;ADDRESS OF RMEC1 FOR TYPEOUT
006210  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMEC1
006214  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006216  104006                          EMT     6
006220  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMEC2
006224  062737  000046  001122          ADD     #RMEC2,$BDADR   ;ADDRESS OF RMEC2 FOR TYPEOUT
006232  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMEC2
006236  001401                          BEQ     .+4             ;CONTENTS ZERO ?
006240  104006                          EMT     6
006242  000004                  5$:     SCOPE                   ;LOOP ?
268
286
287
                                ;;*****************************************************************
                                ;*TEST 3         PORT 'B' SEIZE/TIMEOUT TEST
                                ;*
                                ;*VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT
                                ;*      IT CAN BE RELEASED BY THE ONE SECOND TIMER.
                                ;*
                                ;*  A.  WRITE 0'S INTO RMDA THROUGH PORT 'B';  VERIFY THAT THE DRIVE
                                ;*      HAS BEEN SEIZED.
                                ;*
                                ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'A';
                                ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
                                ;*
                                ;*  C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
                                ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
                                ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
                                ;*      NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS >500 MS.
                                ;*
                                ;;*****************************************************************
006244                          TST3:
006244  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
006250  001406                          BEQ     2$              ;BR IF NOT
006252  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
006254  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
006260  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
006266  012737  006302  001106  2$:     MOV     #TEST3,$LPADR   ;SETUP SCOPE LOOP ADDRESS
006274  012737  006302  001110          MOV     #TEST3,$LPERR   ;SETUP ERROR LOOP ADDRESS
006302                          TEST3:
```

```
        006302  112737  000003  001102          MOVB    #3,$TSTNM       ;MOVE #3 TO TEST NUMBER
        006310  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
        006314  012737  000002  001176          MOV     #2.,$TIMES      ;;DO 2. ITERATIONS
    288
    289 006322  012737  000240  177776          MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
        006330  005037  001270                  CLR     TIMEB           ;CLEAR TIMEOUT VALUE FOR PORT B
        006334  005037  001272                  CLR     TIMEBP  ;CLEAR UPPER TIMEOUT TOLERANCE
        006340  005037  001274                  CLR     TIMEBM  ;CLEAR LOWER TIMEOUT TOLERANCE

                                        ;START THE TIMER

        006344  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
        006350  012737  003720  001260          MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS

                                        ;SEIZE THE DRIVE THROUGH PORT B

        006356  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
        006364  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
        006372  005060  000006                  CLR     RMDA(R0)        ;WRITE RMDA
        006376  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
        006404  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        006412  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
        006420  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
        006426  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
        006432  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
        006440  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
        006444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
        006452  001403                          BEQ     64$             ;BR IF IT IS
        006454  104004                          EMT     4
        006456  000137  007610                  JMP     5$              ;BYPASS REST OF THE SUBTEST
        006462                          64$:
        006462  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
        006470  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        006476  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
        006504  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
        006512  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;EXPECTED STATUS
        006520  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
        006526  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
        006532  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
        006540  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
        006546  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
        006554  001401                          BEQ     65$             ;BR IF THEY ARE
        006556  104005                          EMT     5
        006560  000240                  65$:    NOP

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

        006562  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
        006570  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        006576  016046  000046                  MOV     RMEC2(R0),-(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
        006602  016046  000044                  MOV     RMEC1(R0),-(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
        006606  016046  000030                  MOV     RMSN(R0),-(SP)  ;STORE REGISTER RMSN, PORT A, FOR CHECK
        006612  016046  000034                  MOV     RMDC(R0),-(SP)  ;STORE REGISTER RMDC, PORT A, FOR CHECK
        006616  016046  000032                  MOV     RMOF(R0),-(SP)  ;STORE REGISTER RMOF, PORT A, FOR CHECK
        006622  016046  000042                  MOV     RMER2(R0),-(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
        006626  016046  000020                  MOV     RMLA(R0),-(SP)  ;STORE REGISTER RMLA, PORT A, FOR CHECK
        006632  016046  000026                  MOV     RMDT(R0),-(SP)  ;STORE REGISTER RMDT, PORT A, FOR CHECK
```

```
006636  016046  000006                  MOV     RMDA(R0),-(SP)  ;STORE REGISTER RMDA, PORT A, FOR CHECK
006642  016046  000024                  MOV     RMMR1(R0),-(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
006646  016046  000014                  MOV     RMER1(R0),-(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK

                                ;WAIT FOR PORT B TO TIMEOUT

006652  005760  000012          1$:     TST     RMDS(R0)        ;WAIT FOR THE DRIVE TO TIMEOUT
006656  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
006660  005737  001260                  TST     WATCH           ;CHECK WATCH
006664  001372                          BNE     1$              ;BR IF NOT ZERO
006666  104036                          EMT     36
006670  000137  007274                  JMP     4$              ;BYPASS TIMEOUT TIME CHECK
006674  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS   ;SET PRIORITY TO 7 TO STOP CLOCK
006702  013737  001256  001270          MOV     TIME,TIMEB      ;SAVE THE ELAPSED TIME FOR PORT B
006710  004537  066254                  JSR     R5,TOLER        ;CALCULATE THE TOLERANCE
006714  001270                          .WORD   TIMEB           ;TIMEOUT VALUE FOR PORT B
006716  012637  001272                  MOV     (SP)+,TIMEBP    ;+25% TOLERANCE
006722  012637  001274                  MOV     (SP)+,TIMEBM    ;-25% TOLERANCE

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

006726  023727  001256  000764          CMP     TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
006734  103001                          BHIS    3$              ;BR IF IT  WAS
006736  104055                          EMT     55

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

006740  012737  000240  177776  3$:     MOV     #<5*32.>,@#PS   ;RESTORE PRIORITY TO 5

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

006746  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
006752  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
006760  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
006764  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
006772  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
007000  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
007006  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
007014  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
007022  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007030  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
007036  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
007044  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DON' CARES
007052  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
007060  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007066  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
007074  001006                          BNE     66$             ;BR IF NOT
007076  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
007102  001037                          BNE     68$             ;BR IF NOT
007104  104046                          EMT     46
007106  000137  007272                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
007112  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
007120  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007126  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
007134  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
007140  001414                          BEQ     67$             ;BR IF ZERO
007142  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
007150  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
007156  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
007164  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
007170  001004                          BNE     68$              ;BR IF NOT
007172  012737  177777  001254  67$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
007200  104022                          EMT     22
007202  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
007210  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
007216  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
007224  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
007232  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
007234  104007                          EMT     7
007236  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
007244  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
007252  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
007260  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
007266  001401                          BEQ     70$              ;BR IF OK
007270  104007                          EMT     7
007272  000240                  70$:    NOP

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

007274  013737  001224  001240  4$:     MOV     PORTA,PTNBR      ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007302  010037  001122                  MOV     R0,$BLADR        ;BASE ADDRESS FOR REGISTER RMER1
007306  062737  000014  001122          ADD     #RMER1,$BDADR    ;ADDRESS OF RMER1 FOR TYPEOUT
007314  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMER1
007320  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007322  104006                          EMT     6
007324  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMMR1
007330  062737  000024  001122          ADD     #RMMR1,$BDADR    ;ADDRESS OF RMMR1 FOR TYPEOUT
007336  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMMR1
007342  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007344  104006                          EMT     6
007346  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMDA
007352  062737  000006  001122          ADD     #RMDA,$BDADR     ;ADDRESS OF RMDA FOR TYPEOUT
007360  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMDA
007364  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007366  104006                          EMT     6
007370  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMDT
007374  062737  000026  001122          ADD     #RMDT,$BDADR     ;ADDRESS OF RMDT FOR TYPEOUT
007402  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMDT
007406  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007410  104006                          EMT     6
007412  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMLA
007416  062737  000020  001122          ADD     #RMLA,$BDADR     ;ADDRESS OF RMLA FOR TYPEOUT
007424  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMLA
007430  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007432  104006                          EMT     6
007434  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMER2
007440  062737  000042  001122          ADD     #RMER2,$BDADR    ;ADDRESS OF RMER2 FOR TYPEOUT
007446  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMER2
007452  001401                          BEQ     .+4              ;CONTENTS ZERO ?
007454  104006                          EMT     6
007456  010037  001122                  MOV     R0,$BDADR        ;BASE ADDRESS FOR REGISTER RMOF
007462  062737  000032  001122          ADD     #RMOF,$BDADR     ;ADDRESS OF RMOF FOR TYPEOUT
007470  012637  001126                  MOV     (SP)+,$BDDAT     ;CHECK THE STORED CONTENTS OF RMOF
```

```
007474  001401                          BEQ     .+4             ;CONTENTS ZERO ?
007476  104006                          EMT     6
007500  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMDC
007504  062737  000034  001122          ADD     #RMDC,$BDADR    ;ADDRESS OF RMDC FOR TYPEOUT
007512  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMDC
007516  001401                          BEQ     .+4             ;CONTENTS ZERO ?
007520  104006                          EMT     6
007522  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMSN
007526  062737  000030  001122          ADD     #RMSN,$BDADR    ;ADDRESS OF RMSN FOR TYPEOUT
007534  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMSN
007540  001401                          BEQ     .+4             ;CONTENTS ZERO ?
007542  104006                          EMT     6
007544  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMEC1
007550  062737  000044  001122          ADD     #RMEC1,$BDADR   ;ADDRESS OF RMEC1 FOR TYPEOUT
007556  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMEC1
007562  001401                          BEQ     .+4             ;CONTENTS ZERO ?
007564  104006                          EMT     6
007566  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RMEC2
007572  062737  000046  001122          ADD     #RMEC2,$BDADR   ;ADDRESS OF RMEC2 FOR TYPEOUT
007600  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RMEC2
007604  001401                          BEQ     .+4             ;CONTENTS ZERO ?
007606  104006                          EMT     6
007610  000004                  5$:     SCOPE                   ;LOOP ?
290
304
305
                                ;;*****************************************************************
                                ;*TEST 4         PORT 'A' SEIZE/RELEASE TEST
                                ;*
                                ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  SET VOLUME VALID AND CLEAR ANY ERROR
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT'A'.  VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
                                ;*      DRIVE.
                                ;*
                                ;;*****************************************************************
007612                          TST4:
007612  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
007616  001406                          BEQ     2$              ;BR IF NOT
007620  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
007622  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
007626  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
007634  012737  007650  001106  2$:     MOV     #TEST4,$LPADR   ;SETUP SCOPE LOOP ADDRESS
007642  012737  007650  001110          MOV     #TEST4,$LPERR   ;SETUP ERROR LOOP ADDRESS
007650                          TEST4:
007650  112737  000004  001102          MOVB    #4,$TSTNM       ;MOVE #4 TO TEST NUMBER
007656  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
007662  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
306
339
                                ;START THE TIMER

007670  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
007674  012737  003720  001260          MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS
```

```
                                        ;SEIZE THE DRIVE AND SET VOLUME VALID

                                        ;SEIZE THE DRIVE THROUGH PORT A

007702  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
007710  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
007716  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
007722  013737  001226  001244          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
007730  012760  000021  000000          MOV     #21,RMCS1(R0)           ;SET VOLUME VALID
007736  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
007742  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
007750  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
007756  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
007762  012737  000100  001124          MOV     #VV,$GDDAT  ;WHAT REGISTER SHOULD BE
007770  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
007776  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
010004  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010012  001414                          BEQ     66$             ;BR IF OK
010014  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
010022  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
010030  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
010036  104013                          EMT     13
010040  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
010044  000240                  66$:    NOP
010046  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR DRIVE

                                        ;RELEASE THE DRIVE FROM PORT A

010054  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
010062  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010070  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

010076  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
010102  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
010110  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
010114  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
010122  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
010130  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
010136  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
010144  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
010152  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010160  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
010166  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
010174  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
010202  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
010210  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010216  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
010224  001006                          BNE     68$             ;BR IF NOT
010226  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
010232  001037                          BNE     70$             ;BR IF NOT
010234  104046                          EMT     46
010236  000137  010422                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
010242  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
010250  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
010256  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
010264  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
010270  001414                          BEQ     69$             ;BR IF ZERO
010272  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010300  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
010306  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
010314  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
010320  001004                          BNE     70$             ;BR IF NOT
010322  012737  177777  001254  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
010330  104022                          EMT     22
010332  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
010340  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
010346  042737  100100  001126          BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
010354  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
010362  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
010364  104007                          EMT     7
010366  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
010374  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
010402  042737  100100  001126          BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
010410  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
010416  001401                          BEQ     72$             ;BR IF OK
010420  104007                          EMT     7
010422  000240                  72$:    NOP
010424  005737  001254                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
010430  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
010432  000137  010706                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
010436  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
010444  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010452  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010456  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
010464  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010472  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
010476  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
010502  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010510  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
010516  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010524  001414                          BEQ     73$             ;BR IF OK
010526  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
010534  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
010542  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
010550  104017                          EMT     17
010552  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
010556  000240                  73$:    NOP
010560  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
010566  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010574  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010600  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
010606  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010614  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
010620  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
010624  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010632  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
010640  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010646  001414                          BEQ     75$             ;BR IF OK
010650  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
010656  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
010664  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
010672  104017                          EMT     17
010674  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
010700  000240                  75$:    NOP
010702  000137  010740                  JMP     2$              ;GO CHECK FOR LOOP ON ERROR

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

010706                          1$:
010706  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
010714  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010722  005760  000012                  TST     RMDS(R0)        ;WAIT FOR TIMEOUT TO RELEASE DRIVE
010726  001004                          BNE     2$              ;BR WHEN DRIVE RELEASED
010730  005737  001260                  TST     WATCH           ;CHECK THE WATCH
010734  001364                          BNE     1$              ;BR IF NOT ZERO
010736  104036                          EMT     36
010740  000004                  2$:     SCOPE                   ;LOOP ?
340
354
355                             ;;*****************************************************************
                                ;*TEST 5         PORT 'B' SEIZE/RELEASE TEST
                                ;*
                                ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  SET VOLUME VALID AND CLEAR ANY ERROR
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
                                ;*      DRIVE.
                                ;*
                                ;;*****************************************************************
010742                          TST5:
010742  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
010746  001406                          BEQ     2$              ;BR IF NOT
010750  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
010752  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
010756  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
010764  012737  011000  001106  2$:     MOV     #TEST5,$LPADR   ;SETUP SCOPE LOOP ADDRESS
010772  012737  011000  001110          MOV     #TEST5,$LPERR   ;SETUP ERROR LOOP ADDRESS
011000                          TEST5:
011000  112737  000005  001102          MOVB    #5,$TSTNM       ;MOVE #5 TO TEST NUMBER
011006  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
011012  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
356
357
                                ;START THE TIMER

011020  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
011024  012737  003720  001260          MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS

                                ;SEIZE THE DRIVE AND SET VOLUME VALID

                                ;SEIZE THE DRIVE THROUGH PORT B

011032  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
```

```
011040  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
011046  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
011052  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
011060  012760  000021  000000          MOV     #21,RMCS1(R0)           ;SET VOLUME VALID
011066  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
011072  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
011100  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011106  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
011112  012737  000100  001124          MOV     #VV,$GDDAT  ;WHAT REGISTER SHOULD BE
011120  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
011126  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
011134  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
011142  001414                          BEQ     66$             ;BR IF OK
011144  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
011152  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
011160  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
011166  104013                          EMT     13
011170  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
011174  000240                  66$:    NOP
011176  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR DRIVE

                                ;RELEASE THE DRIVE FROM PORT B

011204  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
011212  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011220  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

011226  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
011232  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
011240  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
011244  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
011252  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
011260  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
011266  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
011274  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
011302  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011310  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
011316  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
011324  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
011332  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
011340  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011346  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
011354  001006                          BNE     68$             ;BR IF NOT
011356  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
011362  001037                          BNE     70$             ;BR IF NOT
011364  104046                          EMT     46
011366  000137  011552                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
011372  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
011400  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011406  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
011414  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
011420  001414                          BEQ     69$             ;BR IF ZERO
011422  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011430  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
011436  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
```

```
011444  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
011450  001004                          BNE     70$             ;BR IF NOT
011452  012737  177777  001254  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
011460  104022                          EMT     22
011462  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
011470  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
011476  042737  100100  001126          BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
011504  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
011512  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
011514  104007                          EMT     7
011516  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
011524  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
011532  042737  100100  001126          BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
011540  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
011546  001401                          BEQ     72$             ;BR IF OK
011550  104007                          EMT     7
011552  000240                  72$:    NOP
011554  005737  001254                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
011560  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
011562  000137  012036                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
011566  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
011574  013737  001226  001240          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011602  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
011606  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
011614  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011622  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
011626  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
011632  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
011640  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
011646  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
011654  001414                          BEQ     73$             ;BR IF OK
011656  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
011664  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
011672  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
011700  104017                          EMT     17
011702  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
011706  000240                  73$:    NOP
011710  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
011716  013737  001224  001240          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011724  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
011730  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
011736  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011744  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
011750  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
011754  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
011762  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
011770  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
011776  001414                          BEQ     75$             ;BR IF OK
012000  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
012006  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
012014  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
012022  104017                          EMT     17
012024  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
012030  000240                  75$:    NOP
012032  000137  012070                  JMP     2$              ;GO CHECK FOR LOOP ON ERROR

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

```
                                        ;TO RELEASE THE DRIVE

     012036                             1$:
     012036  113760  001224  000010            MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
     012044  013737  001224  001240            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     012052  005760  000012                    TST     RMDS(R0)         ;WAIT FOR TIMEOUT TO RELEASE DRIVE
     012056  001004                            BNE     2$               ;BR WHEN DRIVE RELEASED
     012060  005737  001260                    TST     WATCH            ;CHECK THE WATCH
     012064  001364                            BNE     1$               ;BR IF NOT ZERO
     012066  104036                            EMT     36
     012070  000004                     2$:    SCOPE                    ;LOOP ?
 362
 371
 372                                    ;;*****************************************************************
                                        ;*TEST 6        PORT 'A' NEUTRAL/RELEASE TEST
                                        ;*
                                        ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
                                        ;*
                                        ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
                                        ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
                                        ;*
                                        ;;*****************************************************************
     012072                             TST6:
     012072  005737  001300                    TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
     012076  001406                            BEQ     2$               ;BR IF NOT
     012100  100002                            BPL     1$               ;BR IF JUST ENTERED TEST
     012102  000137  003062                    JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
     012106  012737  177777  001300     1$:    MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
     012114  012737  012130  001106     2$:    MOV     #TEST6,$LPADR    ;SETUP SCOPE LOOP ADDRESS
     012122  012737  012130  001110            MOV     #TEST6,$LPERR    ;SETUP ERROR LOOP ADDRESS
     012130                             TEST6:
     012130  112737  000006  001102            MOVB    #6,$TSTNM        ;MOVE #6 TO TEST NUMBER
     012136  012706  001100                    MOV     #STACK,SP        ;LOAD THE STACK POINTER
     012142  012737  000012  001176            MOV     #10.,$TIMES      ;;DO 10. ITERATIONS
 373
 384 012150  113760  001224  000010            MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
     012156  013737  001224  001240            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     012164  013737  001224  001242            MOV     PORTA,SEIZPT     ;ADDR OF PORT WHICH WILL ISSUE RELEASE

                                        ;ISSUE A RELEASE COMMAND
     012172  012760  000013  000000            MOV     #13,RMCS1(R0)    ;ISSUE A RELEASE COMMAND

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

     012200  005037  001254                    CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
     012204  012737  000012  001122            MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
     012212  060037  001122                    ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
     012216  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
     012224  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
     012232  016037  000012  001170            MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
     012240  042737  024001  001170            BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
     012246  013737  001170  001164            MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
     012254  042737  100100  001164            BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
     012262  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
     012270  016037  000012  001172            MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
     012276  042737  024001  001172            BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
     012304  013737  001172  001166            MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
```

```
012312  042737  100100  001166          BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012320  023737  001164  001166          CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
012326  001006                          BNE     64$               ;BR IF NOT
012330  005737  001164                  TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
012334  001045                          BNE     66$               ;BR IF NOT
012336  104046                          EMT     46
012340  000137  012540                  JMP     68$               ;BYPASS THE REST OF THE CHECKS
012344  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012352  013737  001226  001240          MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012360  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
012366  005737  001164                  TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
012372  001414                          BEQ     65$               ;BR IF ZERO
012374  013737  001224  001240          MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012402  013737  001172  001126          MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
012410  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
012416  005737  001166                  TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
012422  001012                          BNE     66$               ;BR IF NOT
012424  012737  177777  001254  65$:    MOV     #-1,RELERR        ;SET 'RELEASE ERROR' INDICATOR
012432  012760  000011  000000          MOV     #11,RMCS1(R0)     ;CLEAR THE DRIVE
012440  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
012446  104030                          EMT     30
012450  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
012456  013737  001224  001240          MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
012464  042737  100000  001126          BIC     #ATA,$BDDAT       ;DON'T CHECK THE ATTN BIT
012472  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;ALL BITS OK ?
012500  001401                          BEQ     67$               ;BR IF OK FROM PORT A.
012502  104007                          EMT     7
012504  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
012512  013737  001226  001240          MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
012520  042737  100000  001126          BIC     #ATA,$BDDAT       ;DON'T CHECK THE ATTN BIT
012526  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;SEE IF READ OK FROM PORT B.
012534  001401                          BEQ     68$               ;BR IF OK
012536  104007                          EMT     7
012540  000240                  68$:    NOP
012542  000004                          SCOPE                     ;LOOP ?
385
394
395                             ;;*****************************************************************
                                ;*TEST 7         PORT 'B' NEUTRAL/RELEASE TEST
                                ;*
                                ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
                                ;*
                                ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
                                ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
                                ;*
                                ;;*****************************************************************
012544                          TST7:
012544  005737  001300                  TST     KYBCTL            ;PERFORMING ONLY SINGLE TEST ?
012550  001406                          BEQ     2$                ;BR IF NOT
012552  100002                          BPL     1$                ;BR IF JUST ENTERED TEST
012554  000137  003062                  JMP     EXEC              ;RETURN & GET NEXT TEST NUMBER
012560  012737  177777  001300  1$:     MOV     #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
012566  012737  012602  001106  2$:     MOV     #TEST7,$LPADR     ;SETUP SCOPE LOOP ADDRESS
012574  012737  012602  001110          MOV     #TEST7,$LPERR     ;SETUP ERROR LOOP ADDRESS
012602                          TEST7:
012602  112737  000007  001102          MOVB    #7,$TSTNM         ;MOVE #7 TO TEST NUMBER
012610  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
```

```
012614  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
396
397 012622  113760  001226  000010      MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
    012630  013737  001226  001240      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    012636  013737  001226  001242      MOV     PORTB,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE

                                ;ISSUE A RELEASE COMMAND
    012644  012760  000013  000000      MOV     #13,RMCS1(R0)   ;ISSUE A RELEASE COMMAND

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

    012652  005037  001254              CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
    012656  012737  000012  001122      MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
    012664  060037  001122              ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
    012670  012737  011700  001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
    012676  113760  001224  000010      MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
    012704  016037  000012  001170      MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
    012712  042737  024001  001170      BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
    012720  013737  001170  001164      MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
    012726  042737  100100  001164      BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
    012734  113760  001226  000010      MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
    012742  016037  000012  001172      MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
    012750  042737  024001  001172      BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
    012756  013737  001172  001166      MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
    012764  042737  100100  001166      BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
    012772  023737  001164  001166      CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
    013000  001006                      BNE     64$             ;BR IF NOT
    013002  005737  001164              TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
    013006  001045                      BNE     66$             ;BR IF NOT
    013010  104046                      EMT     46
    013012  000137  013212              JMP     68$             ;BYPASS THE REST OF THE CHECKS
    013016  013737  001170  001126  64$:  MOV   $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
    013024  013737  001226  001240      MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    013032  113760  001226  000010      MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
    013040  005737  001164              TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
    013044  001414                      BEQ     65$             ;BR IF ZERO
    013046  013737  001224  001240      MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    013054  013737  001172  001126      MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
    013062  113760  001224  000010      MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
    013070  005737  001166              TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
    013074  001012                      BNE     66$             ;BR IF NOT
    013076  012737  177777  001254  65$:  MOV   #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
    013104  012760  000011  000000      MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
    013112  012760  000013  000000      MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
    013120  104030                      EMT     30
    013122  013737  001170  001126  66$:  MOV   $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
    013130  013737  001224  001240      MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
    013136  042737  100000  001126      BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
    013144  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
    013152  001401                      BEQ     67$             ;BR IF OK FROM PORT A.
    013154  104007                      EMT     7
    013156  013737  001172  001126  67$:  MOV   $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
    013164  013737  001226  001240      MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
    013172  042737  100000  001126      BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
    013200  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
    013206  001401                      BEQ     68$             ;BR IF OK
    013210  104007                      EMT     7
```

```
        013212  000240                    68$:    NOP
        013214  000004                            SCOPE                   ;LOOP ?
    398
    417
    418                                   ;;*****************************************************************
                                          ;*TEST 10        PORT 'A' RELEASE INTERFERENCE TEST
                                          ;*
                                          ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
                                          ;*      IS SEIZED BY THE OTHER PORT.
                                          ;*
                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                          ;*
                                          ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
                                          ;*
                                          ;*  C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
                                          ;*
                                          ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE SWITCHED
                                          ;*      TO PORT 'A'.
                                          ;*
                                          ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
                                          ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                          ;*
                                          ;;*****************************************************************
        013216                            TST10:
        013216  005737  001300                    TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
        013222  001406                            BEQ     2$              ;BR IF NOT
        013224  100002                            BPL     1$              ;BR IF JUST ENTERED TEST
        013226  000137  003062                    JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
        013232  012737  177777  001300    1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
        013240  012737  013254  001106    2$:     MOV     #TEST10,$LPADR  ;SETUP SCOPE LOOP ADDRESS
        013246  012737  013254  001110            MOV     #TEST10,$LPERR  ;SETUP ERROR LOOP ADDRESS
        013254                            TEST10:
        013254  112737  000010  001102            MOVB    #10,$TSTNM      ;MOVE #10 TO TEST NUMBER
        013262  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
        013266  012737  000012  001176            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
    419
    443
                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS

        013274  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
        013302  005060  000012                    CLR     RMDS(R0)        ;SEIZE THE DRIVE
        013306  012760  000011  000000            MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
        013314  012760  000013  000000            MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
        013322  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
        013330  005060  000012                    CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
        013334  012760  000011  000000            MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
        013342  012760  000013  000000            MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                          ;SEIZE THE DRIVE THROUGH PORT B

        013350  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
        013356  013737  001226  001242            MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
        013364  005060  000012                    CLR     RMDS(R0)        ;WRITE RMDS
        013370  113760  001224  000010            MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
        013376  013737  001224  001240            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        013404  013737  001224  001244            MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
        013412  016037  000012  001126            MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
```

```
013420  010037  001122                  MOV     R0,$BDADR          ;RH/RM BASE ADDRESS
013424  062737  000012  001122          ADD     #RMDS,$BDADR       ;GENERATE BAD REGISTER ADDRESS
013432  005037  001124                  CLR     $GDDAT             ;REGISTER SHOULD BE ZERO
013436  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER ZERO
013444  001403                          BEQ     64$                ;BR IF IT IS
013446  104004                          EMT     4
013450  000137  014460                  JMP     1$                 ;BYPASS REST OF THE SUBTEST
013454                          64$:
013454  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
013462  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013470  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
013476  042737  020001  001126          BIC     #OM!P!P,$BDDAT          ;CLEAR DONT CARE BITS
013504  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
013512  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
013520  005137  001166                  COM     $TMP1              ;COMPLEMENT THE EXPECTED STATUS
013524  013737  001126  001164          MOV     $BDDAT,$TMP0       ;SAVE THE ACTUAL STATUS
013532  043737  001166  001164          BIC     $TMP1,$TMP0        ;CLEAR UNWANTED BITS
013540  023737  001124  001164          CMP     $GDDAT,$TMP0       ;ARE THE EXPECTED STATUS BITS SET ?
013546  001401                          BEQ     65$                ;BR IF THEY ARE
013550  104005                          EMT     5
013552  000240                  65$:    NOP

                                ;TRY TO FXECUTE A RELEASE COMMAND THROUGH PORT A

013554  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
013562  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013570  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE A RELEASE COMMAND THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

013576  005037  001250                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
013602  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
013610  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
013616  060037  001122                  ADD     R0,$BDADR          ;ADD RH/RM BASE ADDRESS
013622  005037  001124                  CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
013626  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
013634  001403                          BEQ     66$                ;BR IF OK
013636  104010                          EMT     10
013640  005137  001250                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
013644  016037  000000  001126  66$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
013652  012737  000000  001122          MOV     #RMCS1,$BDADR      ;FORM ADDRESS OF REGISTER
013660  060037  001122                  ADD     R0,$BDADR          ;ADDRESS BASE
013664  032737  020000  001126          BIT     #MCPE,$BDDAT       ;IS 'MCPE' SET ?
013672  001404                          BEQ     67$                ;BR IF NOT
013674  104011                          EMT     11
013676  012760  040000  000000          MOV     #TRE,RMCS1(R0)   ;CLEAR 'MCPE'
013704  000240                  67$:    NOP
013706  005737  001250                  TST     CKERR              ;WAS RMDS NON ZERO ?
013712  001402                          BEQ     .+6                ;CONTENTS OF RMDS SEEN BY PORT A
013714  000137  014460                  JMP     1$                 ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

                                ;RELEASE THE DRIVE FROM PORT B

013720  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
013726  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013734  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B
```

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

013742  005037  001254                          CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
013746  012737  111700  001124                  MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
013754  012737  000012  001122                  MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
013762  060037  001122                          ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
013766  113760  001224  000010                  MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
013774  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014002  016037  000012  001164                  MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT A
014010  113760  001226  000010                  MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
014016  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014024  016037  000012  001126                  MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
014032  001404                                  BEQ     68$             ;BR IF STATUS FROM PORT B ZERO
014034  005737  001164                          TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
014040  001401                                  BEQ     68$             ;BR IF ZERO
014042  104031                                  EMT     31
014044  013737  001164  001126  68$:            MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
014052  013737  001224  001240                  MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
014060  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
014066  001401                                  BEQ     69$             ;BR IF OK
014070  104027                                  EMT     27
014072  000240                          69$:    NOP

                                        ;RELEASE THE DRIVE FROM PORT A

014074  113760  001224  000010                  MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
014102  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014110  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

014116  005037  001254                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
014122  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
014130  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
014134  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
014142  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
014150  016037  000012  001170                  MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
014156  042737  024001  001170                  BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
014164  013737  001170  001164                  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
014172  042737  100100  001164                  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014200  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
014206  016037  000012  001172                  MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
014214  042737  024001  001172                  BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
014222  013737  001172  001166                  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
014230  042737  100100  001166                  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014236  023737  001164  001166                  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
014244  001006                                  BNE     70$             ;BR IF NOT
014246  005737  001164                          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
014252  001045                                  BNE     72$             ;BR IF NOT
014254  104046                                  EMT     46
014256  000137  014456                          JMP     74$             ;BYPASS THE REST OF THE CHECKS
014262  013737  001170  001126  70$:            MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014270  013737  001226  001240                  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014276  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
014304  005737  001164                          TST     $TMP0           ;SEE IF STATUS EQ 0 FRO. PORT A.
014310  001414                                  BEQ     71$             ;BR IF ZERO
014312  013737  001224  001240                  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
014320  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
014326  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
014334  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
014340  001012                          BNE     72$              ;BR IF NOT
014342  012737  177777  001254  71$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
014350  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
014356  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
014364  104026                          EMT     26
014366  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
014374  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
014402  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
014410  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
014416  001401                          BEQ     73$              ;BR IF OK FROM PORT A.
014420  104007                          EMT     7
014422  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
01443C  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
014436  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
014444  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
014452  001401                          BEQ     74$              ;BR IF OK
014454  104007                          EMT     7
014456  000240                  74$:    NOP
014460  000004                  1$:     SCOPE                    ;LOOP ?
```

```
                                ;;*****************************************************************
                                ;*TEST 11        PORT 'B' RELEASE INTERFERENCE TEST
                                ;*
                                ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
                                ;*      IS SEIZED BY THE OTHER PORT.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  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.
                                ;*
                                ;;*****************************************************************
014462                          TST11:
014462  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
014466  001406                          BEQ     2$               ;BR IF NOT
014470  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
014472  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
014476  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
014504  012737  014520  001106  2$:     MOV     #TEST11,$LPADR   ;SETUP SCOPE LOOP ADDRESS
014512  012737  014520  001110          MOV     #TEST11,$LPERR   ;SETUP ERROR LOOP ADDRESS
014520                          TEST11:
014520  112737  000011  001102          MOVB    #11,$TSTNM       ;MOVE #11 TO TEST NUMBER
014526  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
014532  012737  000012  001176          MOV     #10.,$TIMES      ;;DO 10. ITERATIONS
```

```
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

014540  113760  001224  000010           MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
014546  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
014552  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
014560  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
014566  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
014574  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
014600  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
014606  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

014614  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
014622  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
014630  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
014634  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
014642  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014650  013737  001226  001244          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
014656  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
014664  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
014670  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
014676  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
014702  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
014710  001403                          BEQ     64$             ;BR IF IT IS
014712  104004                          EMT     4
014714  000137  015724                  JMP     1$              ;BYPASS REST OF THE SUBTEST
014720                          64$:
014720  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
014726  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014734  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
014742  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
014750  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
014756  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
014764  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
014770  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
014776  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
015004  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
015012  001401                          BEQ     65$             ;BR IF THEY ARE
015014  104005                          EMT     5
015016  000240                  65$:    NOP

                                        ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B

015020  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
015026  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015034  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT B

                                        ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A

015042  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
015046  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
015054  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015062  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
015066  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
015072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
015100  001403                          BEQ     66$             ;BR IF OK
```

```
015102  104010                          EMT     10
015104  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
015110  016037  000000  001126  66$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
015116  012737  000000  001122          MOV     #RMCS1,$BDADR   ;FORM ADDRESS OF REGISTER
015124  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
015130  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
015136  001404                          BEQ     67$             ;BR IF NOT
015140  104011                          EMT     11
015142  012760  040000  000000          MOV     #TRE,RMCS1(R0)  ;CLEAR 'MCPE'
015150  000240                  67$:    NOP
015152  005737  001250                  TST     CKERR           ;WAS RMDS NON ZERO ?
015156  001402                          BEQ     .+6             ;CONTENTS OF RMDS SEEN BY PORT B
015160  000137  015724                  JMP     1$              ;DRIVE .N NEUTRAL, BYPASS REST OF TEST

                                ;RELEASE THE DRIVE FROM PORT A

015164  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
015172  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015200  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

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

015206  005037  001254                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
015212  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
015220  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
015226  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
015232  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
015240  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015246  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT B
015254  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
015262  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015270  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
015276  001404                          BEQ     68$             ;BR IF STATUS FROM PORT A ZERO
015300  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
015304  001401                          BEQ     68$             ;BR IF ZERO
015306  104031                          EMT     31
015310  013737  001164  001126  68$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
015316  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
015324  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
015332  001401                          BEQ     69$             ;BR IF OK
015334  104027                          EMT     27
015336  000240                  69$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

015340  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
015346  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015354  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

015362  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
015366  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
015374  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
015400  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
015406  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
015414  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
```

```
015422  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
015430  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
015436  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015444  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
015452  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
015460  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
015466  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
015474  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015502  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
015510  001006                          BNE     70$              ;BR IF NOT
015512  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
015516  001045                          BNE     72$              ;BR IF NOT
015520  104046                          EMT     46
015522  000137  015722                  JMP     74$              ;BYPASS THE REST OF THE CHECKS
015526  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
015534  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015542  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
015550  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
015554  001414                          BEQ     71$              ;BR IF ZERO
015556  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015564  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
015572  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
015600  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
015604  001012                          BNE     72$              ;BR IF NOT
015606  012737  177777  001254  71$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
015614  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
015622  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
015630  104026                          EMT     26
015632  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
015640  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
015646  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
015654  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
015662  001401                          BEQ     73$              ;BR IF OK FROM PORT A.
015664  104007                          EMT     7
015666  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
015674  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
015702  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
015710  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
015716  001401                          BEQ     74$              ;BR IF OK
015720  104007                          EMT     7
015722  000240                  74$:    NOP
015724  000004                  1$:     SCOPE                    ;LOOP ?
467
487
488
                                ;;*****************************************************************
                                ;*TEST 12        PORT 'A' RELEASE W/ERRORS TEST
                                ;*
                                ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
                                ;*      BITS ARE SET IN THE DRIVE.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE 'GO'
                                ;*      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
                                ;*      THAT RMER1 HAS NOT BEEN CLEARED.
```

```
                                        ;*
                                        ;*  D.  CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
                                        ;*
                                        ;*  E.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;*********************************************** ****************
     015726                             TST12:
     015726  005737  001300                     TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
     015732  001406                             BEQ     2$              ;BR IF NOT
     015734  100002                             BPL     1$              ;BR IF JUST ENTERED TEST
     015736  000137  003062                     JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
     015742  012737  177777  001300     1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
     015750  012737  015764  001106     2$:     MOV     #TEST12,$LPADR  ;SETUP SCOPE LOOP ADDRESS
     015756  012737  015764  001110             MOV     #TEST12,$LPERR  ;SETUP ERROR LOOP ADDRESS
     015764                             TEST12:
     015764  112737  000012  001102             MOVB    #12,$TSTNM      ;MOVE #12 TO TEST NUMBER
     015772  012706  001100                     MOV     #STACK,SP       ;LOAD THE STACK POINTER
     015776  012737  000012  001176             MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
489
523
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

     016004  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
     016012  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE
     016016  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     016024  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
     016032  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
     016040  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
     016044  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     016052  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

     016060  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
     016066  013737  001224  001242             MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
     016074  005060  000012                     CLR     RMDS(R0)        ;WRITE RMDS
     016100  013737  001226  001244             MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;FORCE AN ERROR

     016106  012760  177777  000014             MOV     #-1,RMER1(R0)   ;SET ERROR BITS
     016114  012760  000013  000000             MOV     #13,RMCS1(R0)   ;ISSUE A RELEASE COMMAND
     016122  005037  001250                     CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
     016126  016037  000000  001126             MOV     RMCS1(R0),$BDDAT ;GET CONTENTS OF RMCS1
     016134  012737  000000  001122             MOV     #RMCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
     016142  060037  001122                     ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
     016146  012737  004012  001124             MOV     #4012,$GDDAT    ;WHAT REGISTER SHOULD BE
     016154  013737  001126  001164             MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
     016162  042737  173765  001164             BIC     #^C4012,$TMP0   ;SAVE SPECIFIED BITS
     016170  023737  001124  001164             CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
     016176  001414                             BEQ     66$             ;BR IF OK
     016200  013737  001126  001174             MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
     016206  042737  004012  001174             BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
     016214  053737  001174  001124             BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
     016222  104025                             EMT     25
     016224  005137  001250                     COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
016230  000240                          66$:    NOP
016232  005737  001250                          TST     CKERR           ;DID 'GO' BIT RESET ?
016236  001002                                  BNE     .+6             ;BR IF NOT
016240  000137  016300                          JMP     1$              ;'GO' BIT RESET
016244  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;INIT THE RH/RM
016252  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
016260  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016266  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT A
016274  000137  017044                          JMP     2$              ;BYPASS THE REST OF THE TEST

                                        ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A

016300                                  1$:
016300  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
016306  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016314  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
016320  016037  000012  001126                  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
016326  012737  000012  001122                  MOV     #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016334  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
016340  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
016344  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
016352  001403                                  BEQ     68$             ;BR IF OK
016354  104024                                  EMT     24
016356  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
016362  000240                          68$:    NOP
016364  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
016372  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016400  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
016404  016037  000014  001126                  MOV     RMER1(R0),$BDDAT ;GET CONTENTS OF RMER1
016412  012737  000014  001122                  MOV     #RMER1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016420  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
016424  012737  177777  001124                  MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
016432  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
016440  001403                                  BEQ     70$             ;BR IF OK
016442  104010                                  EMT     10
016444  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
016450  000240                          70$:    NOP

                                        ;CLEAR THE ERRORS THROUGH PORT A

016452  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE A DRIVE CLEAR

                                        ;RELEASE THE DRIVE FROM PORT A

016460  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
016466  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016474  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

016502  005037  001254                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
016506  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
016514  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
016520  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
016526  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
016534  016037  000012  001170                  MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
016542  042737  024001  001170                  BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
```

```
016550  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
016556  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016564  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
016572  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
016600  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
016606  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
016614  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016622  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
016630  001006                          BNE     72$              ;BR IF NOT
016632  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
016636  001045                          BNE     74$              ;BR IF NOT
016640  104046                          EMT     46
016642  000137  017042                  JMP     76$              ;BYPASS THE REST OF THE CHECKS
016646  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
016654  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016662  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
016670  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
016674  001414                          BEQ     73$              ;BR IF ZERO
016676  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016704  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
016712  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
016720  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
016724  001012                          BNE     74$              ;BR IF NOT
016726  012737  177777  001254  73$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
016734  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
016742  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
016750  104026                          EMT     26
016752  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
016760  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
016766  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
016774  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
017002  001401                          BEQ     75$              ;BR IF OK FROM PORT A.
017004  104007                          EMT     7
017006  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
017014  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
017022  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
017030  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
017036  001401                          BEQ     76$              ;BR IF OK
017040  104007                          EMT     7
017042  000240                  76$:    NOP
017044  000004                  2$:     SCOPE                    ;LOOP ?
524
544
545
                                ;;*****************************************************************
                                ;*TEST 13        PORT 'B' RELEASE W/ERRORS TEST
                                ;*
                                ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
                                ;*      BITS ARE SET IN THE DRIVE.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WRITE 1'S INTO RMER1 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 RMER1 HAS NOT BEEN CLEARED.
                                ;*
```

```
                                        ;*  D.  CLEAR RMER1 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.
                                        ;*
                                        ;*****************************************************************
017046                                  TST13:
017046  005737  001300                          TST     KVBCTL          ;PERFORMING ONLY SINGLE TEST ?
017052  001406                                  BEQ     2$              ;BR IF NOT
017054  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
017056  000137  003062                          JMP     EXEC                TURN & GET NEXT TEST NUMBER
017062  012737  177777  001300          1$:     MOV     #-1,KVBCTL          T SINGLE TEST INDICATOR
017070  012737  017104  001106          2$:     MOV     #TEST13,$LPAL       SETUP SCOPE LOOP ADDRESS
017076  012737  017104  001110                  MOV     #TEST13,$LPERR      SETUP ERROR LOOP ADDRESS
017104                                  TEST13:
017104  112737  000013  001102                  MOVB    #13,$TSTNR      ;MOVE #13 TO TEST NUMBER
017112  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
017116  012737  000012  001176                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

017124  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
017132  005060  000012                          CLR     RMDS(R0)        ;SEIZE THE DRIVE
017136  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
017144  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
017152  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
017160  005060  000012                          CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
017164  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
017172  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT B

017200  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
017206  013737  001226  001242                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
017214  005060  000012                          CLR     RMDS(R0)        ;WRITE RMDS
017220  013737  001224  001244                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;FORCE AN ERROR

017226  012760  177777  000014                  MOV     #-1,RMER1(R0)   ;SET ERROR BITS
017234  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;ISSUE A RELEASE COMMAND
017242  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
017246  016037  000000  001126                  MOV     RMCS1(R0),$BDDAT ;GET CONTENTS OF RMCS1
017254  012737  000000  001122                  MOV     #RMCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017262  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
017266  012737  004012  001124                  MOV     #4012,$GDDAT    ;WHAT REGISTER SHOULD BE
017274  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
017302  042737  173765  001164                  BIC     #^C4012,$TMP0   ;SAVE SPECIFIED BITS
017310  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
017316  001414                                  BEQ     66$             ;BR IF OK
017320  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
017326  042737  004012  001174                  BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
017334  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
017342  104025                                  EMT     25
017344  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
017350  000240                          66$:    NOP
```

```
017352  005737  001250                  TST     CKERR           ;DID 'GO' BIT RESET ?
017356  001002                          BNE     .+6             ;BR IF NOT
017360  000137  017420                  JMP     1$              ;'GO' BIT RESET
017364  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;INIT THE RH/RM
017372  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
017400  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017406  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT B
017414  000137  020164                  JMP     2$              ;BYPASS THE REST OF THE TEST

                                ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B

017420                          1$:
017420  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
017426  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017434  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
017440  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
017446  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017454  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
017460  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
017464  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
017472  001403                          BEQ     68$             ;BR IF OK
017474  104024                          EMT     24
017476  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
017502  000240                  68$:    NOP
017504  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
017512  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017520  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
017524  016037  000014  001126          MOV     RMER1(R0),$BDDAT  ;GET CONTENTS OF RMER1
017532  012737  000014  001122          MOV     #RMER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017540  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
017544  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
017552  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
017560  001403                          BEQ     70$             ;BR IF OK
017562  104010                          EMT     10
017564  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
017570  000240                  70$:    NOP

                                ;CLEAR THE ERRORS THROUGH PORT B

017572  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE A DRIVE CLEAR

                                ;RELEASE THE DRIVE FROM PORT B

017600  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
017606  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017614  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

017622  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
017626  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
017634  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
017640  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
017646  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
017654  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
017662  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
017670  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
017676  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
017704  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
017712  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
017720  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
017726  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
017734  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
017742  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
017750  001006                          BNE     72$              ;BR IF NOT
017752  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
017756  001045                          BNE     74$              ;BR IF NOT
017760  104046                          EMT     46
017762  000137  020162                  JMP     76$              ;BYPASS THE REST OF THE CHECKS
017766  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
017774  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020002  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
020010  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
020014  001414                          BEQ     73$              ;BR IF ZERO
020016  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020024  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
020032  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
020040  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
020044  001012                          BNE     74$              ;BR IF NOT
020046  012737  177777  001254  73$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
020054  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
020062  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
020070  104026                          EMT     26
020072  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
020100  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
020106  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
020114  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
020122  001401                          BEQ     75$              ;BR IF OK FROM PORT A.
020124  104007                          EMT     7
020126  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
020134  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
020142  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
020150  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
020156  001401                          BEQ     76$              ;BR IF OK
020160  104007                          EMT     7
020162  000240                  76$:    NOP
020164  000004                  2$:     SCOPE                    ;LOOP ?
548
567
568
                                ;;*****************************************************************
                                ;*TEST 14        PORT 'A' SEIZE AND CLEAR TEST
                                ;*
                                ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
                                ;*      PORT TO RELEASE THE DRIVE.
                                ;*
                                ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'A'.
                                ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
                                ;*
                                ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
                                ;*      DOES NOT RETURN TO NEUTRAL.
                                ;*
                                ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH/RM 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.
                                        ;*
                                        ;;*****************************************************************
020166                                  TST14:
020166  005737  001300                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
020172  001406                                  BEQ     2$              ;BR IF NOT
020174  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
020176  000137  003062                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
020202  012737  177777  001300  1$:             MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
020210  012737  020224  001106  2$:             MOV     #TEST14,$LPADR  ;SETUP SCOPE LOOP ADDRESS
020216  012737  020224  001110                  MOV     #TEST14,$LPERR  ;SETUP ERROR LOOP ADDRESS
020224                                  TEST14:
020224  112737  000014  001102                  MOVB    #14,$TSTNM      ;MOVE #14 TO TEST NUMBER
020232  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
020236  012737  000012  001176                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
569
599
                                        ;SEIZE THE DRIVE THROUGH PORT A

020244  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
020252  013737  001224  001242                  MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
020260  005060  000012                          CLR     RMDS(R0)        ;WRITE RMDS
020264  113760  001226  000010                  MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
020272  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020300  013737  001226  001244                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
020306  016037  000012  001126                  MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
020314  010037  001122                          MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
020320  062737  000012  001122                  ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
020326  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
020332  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
020340  001403                                  BEQ     64$             ;BR IF IT IS
020342  104004                                  EMT     4
020344  000137  021564                          JMP     1$              ;BYPASS REST OF THE SUBTEST
020350                                  64$:
020350  113760  001224  000010                  MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
020356  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020364  016037  000012  001126                  MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
020372  042737  020001  001126                  BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
020400  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
020406  013737  001124  001166                  MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
020414  005137  001166                          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
020420  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
020426  043737  001166  001164                  BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
020434  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
020442  001401                                  BEQ     65$             ;BR IF THEY ARE
020444  104005                                  EMT     5
020446  000240                          65$:    NOP

                                        ;DRIVE CLEAR THROUGH PORT A FIRST

020450  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT A

                                        ;VERIFY THAT DRIVE STILL SEIZED BY PORT A

020456  113760  001226  000010                  MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
020464  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
020472  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
020476  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
020504  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020512  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
020516  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
020522  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
020530  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
020536  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
020544  001414                          BEQ     66$             ;BR IF OK
020546  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
020554  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
020562  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
020570  104033                          EMT     33
020572  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
020576  000240                  66$:    NOP
020600  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
020606  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020614  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
020620  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
020626  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020634  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
020640  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
020646  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
020654  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
020662  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
020670  001414                          BEQ     68$             ;BR IF OK
020672  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
020700  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
020706  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
020714  104033                          EMT     33
020716  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
020722  000240                  68$:    NOP

                                ;NOW ISSUE MASSBUS INIT

020724  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;ISSUE MASSBUS INIT

                                ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

020732  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
020740  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020746  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
020752  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
020760  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020766  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
020772  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
020776  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
021004  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
021012  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
021020  001414                          BEQ     70$             ;BR IF OK
021022  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
021030  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
021036  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
021044  104034                          EMT     34
021046  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
021052  000240                  70$:    NOP
021054  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
```

```
021062  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021070  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
021074  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
021102  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021110  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
021114  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
021122  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
021130  042737  100000  001164          BIC     #^C77777,$TMP0   ;SAVE SPECIFIED BITS
021136  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
021144  001414                          BEQ     72$              ;BR IF OK
021146  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
021154  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
021162  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
021170  104034                          EMT     34
021172  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
021176  000240                  72$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

021200  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
021206  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021214  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

021222  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
021226  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
021234  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
021240  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
021246  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
021254  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
021262  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
021270  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
021276  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021304  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
021312  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
021320  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
021326  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
021334  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021342  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
021350  001006                          BNE     74$              ;BR IF NOT
021352  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
021356  001045                          BNE     76$              ;BR IF NOT
021360  104046                          EMT     46
021362  000137  021562                  JMP     78$              ;BYPASS THE REST OF THE CHECKS
021366  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
021374  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021402  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
021410  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
021414  001414                          BEQ     75$              ;BR IF ZERO
021416  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021424  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
021432  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
021440  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
021444  001012                          BNE     76$              ;BR IF NOT
021446  012737  177777  001254  75$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
021454  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
```

```
021462  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
021470  104026                          EMT     26
021472  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
021500  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
021506  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
021514  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
021522  001401                          BEQ     77$              ;BR IF OK FROM PORT A.
021524  104007                          EMT     7
021526  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
021534  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
021542  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
021550  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
021556  001401                          BEQ     78$              ;BR IF OK
021560  104007                          EMT     7
021562  000240                  78$:    NOP
021564  000004                  1$:     SCOPE                    ;LOOP ?
```

```
                                ;;*****************************************************************
                                ;*TEST 15        PORT 'B' SEIZE AND CLEAR TEST
                                ;*
                                ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
                                ;*      PORT TO RELEASE THE DRIVE.
                                ;*
                                ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'B'.
                                ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
                                ;*
                                ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
                                ;*      DOES NOT RETURN TO NEUTRAL.
                                ;*
                                ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH/RM 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.
                                ;*
                                ;;*****************************************************************
021566                          TST15:
021566  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
021572  001406                          BEQ     2$               ;BR IF NOT
021574  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
021576  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
021602  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
021610  012737  021624  001106  2$:     MOV     #TEST15,$LPADR   ;SETUP SCOPE LOOP ADDRESS
021616  012737  021624  001110          MOV     #TEST15,$LPERR   ;SETUP ERROR LOOP ADDRESS
021624                          TEST15:
021624  112737  000015  001102          MOVB    #15,$TSTNM       ;MOVE #15 TO TEST NUMBER
021632  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
021636  012737  000012  001176          MOV     #10.,$TIMES      ;;DO 10. ITERATIONS

                                ;SEIZE THE DRIVE THROUGH PORT B

021644  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
021652  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
021660  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
021664  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
```

```
021672  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021700  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
021706  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
021714  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
021720  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
021726  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
021732  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
021740  001403                          BEQ     64$             ;BR IF IT IS
021742  104004                          EMT     4
021744  000137  023164                  JMP     1$              ;BYPASS REST OF THE SUBTEST
021750                          64$:
021750  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
021756  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021764  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
021772  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
022000  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
022006  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
022014  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
022020  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
022026  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
022034  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
022042  001401                          BEQ     65$             ;BR IF THEY ARE
022044  104005                          EMT     5
022046  000240                  65$:    NOP

                                ;DRIVE CLEAR THROUGH PORT B FIRST

022050  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT B

                                ;VERIFY THAT DRIVE STILL SEIZED BY PORT B

022056  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
022064  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022072  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
022076  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
022104  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022112  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
022116  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
022122  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
022130  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
022136  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
022144  001414                          BEQ     66$             ;BR IF OK
022146  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
022154  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
022162  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
022170  104033                          EMT     33
022172  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
022176  000240                  66$:    NOP
022200  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
022206  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022214  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
022220  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
022226  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022234  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
022240  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
022246  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
022254  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
```

```
022262  023737  001124  001164          CMP     $GDDAT,$TMPO     ;COMPARE THE BITS
022270  001414                          BEQ     68$              ;BR IF OK
022272  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
022300  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
022306  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
022314  104033                          EMT     33
022316  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
022322  000240                  68$:    NOP

                                ;NOW ISSUE MASSBUS INIT

022324  012760  000040  000010          MOV     #CLR,RMCS2(R0)   ;ISSUE MASSBUS INIT

                                ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B

022332  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
022340  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022346  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
022352  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTE.    OF RMDS
022360  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022366  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
022372  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
022376  013737  001126  001164          MOV     $BDDAT,$TMPO     ;MO 'E REGISTER CONTENTS TO '$TMPO'
022404  042737  100000  001164          BIC     #^C77777,$TMPO   ;SAVE SPECIFIED BITS
022412  023737  001124  001164          CMP     $GDDAT,$TMPO     ;COMPARE THE BITS
022420  001414                          BEQ     70$              ;BR IF OK
022422  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
022430  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
022436  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
022444  104034                          EMT     34
022446  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
022452  000240                  70$:    NOP
022454  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
022462  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022470  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
022474  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
022502  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022510  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
022514  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
022522  013737  001126  001164          MOV     $BDDAT,$TMPO     ;MOVE REGISTER CONTENTS TO '$TMPO'
022530  042737  100000  001164          BIC     #^C77777,$TMPO   ;SAVE SPECIFIED BITS
022536  023737  001124  001164          CMP     $GDDAT,$TMPO     ;COMPARE THE BITS
022544  001414                          BEQ     72$              ;BR IF OK
022546  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
022554  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
022562  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
022570  104034                          EMT     34
022572  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
022576  000240                  72$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

022600  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
022606  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022614  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

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

```
022622  005037  001254                 CLR     RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
022626  012737  000012  001122         MOV     #RMDS,$BDADR      ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
022634  060037  001122                 ADD     R0,$BDADR         ;ADD THE I/O BASE ADDRESS
022640  012737  011700  001124         MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
022646  113760  001224  000010         MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
022654  016037  000012  001170         MOV     RMDS(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022662  042737  024001  001170         BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
022670  013737  001170  001164         MOV     $TMP2,$TMP0       ;COPY IT INTO '$TMP0'
022676  042737  100100  001164         BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022704  113760  001226  000010         MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
022712  016037  000012  001172         MOV     RMDS(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022720  042737  024001  001172         BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
022726  013737  001172  001166         MOV     $TMP3,$TMP1       ;COPY IT INTO '$TMP1'
022734  042737  100100  001166         BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022742  023737  001164  001166         CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
022750  001006                         BNE     74$               ;BR IF NOT
022752  005737  001164                 TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
022756  001045                         BNE     76$               ;BR IF NOT
022760  104046                         EMT     46
022762  000137  023162                 JMP     78$               ;BYPASS THE REST OF THE CHECKS
022766  013737  001170  001126  74$:   MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
022774  013737  001226  001240         MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023002  113760  001226  000010         MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
023010  005737  001164                 TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
023014  001414                         BEQ     75$               ;BR IF ZERO
023016  013737  001224  001240         MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023024  013737  001172  001126         MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
023032  113760  001224  000010         MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
023040  005737  001166                 TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
023044  001012                         BNE     76$               ;BR IF NOT
023046  012737  177777  001254  75$:   MOV     #-1,RELERR        ;SET 'RELEASE ERROR' INDICATOR
023054  012760  000011  000000         MOV     #11,RMCS1(R0)     ;CLEAR THE DRIVE
023062  012760  000013  000000         MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
023070  104026                         EMT     26
023072  013737  001170  001126  76$:   MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
023100  013737  001224  001240         MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
023106  042737  100000  001126         BIC     #ATA,$BDDAT       ;DON'T CHECK THE ATTN BIT
023114  023737  001124  001126         CMP     $GDDAT,$BDDAT     ;ALL BITS OK ?
023122  001401                         BEQ     77$               ;BR IF OK FROM PORT A.
023124  104007                         EMT     7
023126  013737  001172  001126  77$:   MOV     $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
023134  013737  001226  001240         MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
023142  042737  100000  001126         BIC     #ATA,$BDDAT       ;DON'T CHECK THE ATTN BIT
023150  023737  001124  001126         CMP     $GDDAT,$BDDAT     ;SEE IF READ OK FROM PORT B.
023156  001401                         BEQ     78$               ;BR IF OK
023160  104007                         EMT     7
023162  000240                  78$:   NOP
023164  000004                  1$:    SCOPE                    ;LOOP ?
623
635
636                             ;;*****************************************************************
                                ;*TEST 16        SEIZE 'A' BY RMCS1 TEST
                                ;*
                                ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
                                ;*IF THE DRIVE IS IN NEUTRAL.
                                ;*  A.  READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'A'; VERIFY THAT
```

```
                                        ;*      THE DRIVE IS SEIZED.
                                        ;*
                                        ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
     023166                             TST16:
     023166  005737  001300                     TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
     023172  001406                             BEQ     2$              ;BR IF NOT
     023174  100002                             BPL     1$              ;BR IF JUST ENTERED TEST
     023176  000137  003062                     JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
     023202  012737  177777  001300     1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
     023210  012737  023224  001106     2$:     MOV     #TEST16,$LPADR  ;SETUP SCOPE LOOP ADDRESS
     023216  012737  023224  001110             MOV     #TEST16,$LPERR  ;SETUP ERROR LOOP ADDRESS
     023224                             TEST16:
     023224  112737  000016  001102             MOVB    #16,$TSTNM      ;MOVE #16 TO TEST NUMBER
     023232  012706  001100                     MOV     #STACK,SP       ;LOAD THE STACK POINTER
     023236  012737  000012  001176             MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
   637
   647
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

     023244  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
     023252  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE
     023256  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     023264  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
     023272  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
     023300  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
     023304  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     023312  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

     023320  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
     023326  013737  001224  001242             MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
     023334  005760  000000                     TST     RMCS1(R0)       ;READ RMCS1
     023340  113760  001226  000010             MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
     023346  013737  001226  001240             MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     023354  013737  001226  001244             MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
     023362  016037  000012  001123             MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
     023370  010037  001122                     MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
     023374  062737  000012  001122             ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
     023402  005037  001124                     CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
     023406  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
     023414  001403                             BEQ     64$             ;BR IF IT IS
     023416  104004                             EMT     4
     023420  000137  024110                     JMP     1$              ;BYPASS REST OF THE SUBTEST
     023424                             64$:
     023424  113760  001224  000010             MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
     023432  013737  001224  001240             MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     023440  016037  000012  001126             MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
     023446  042737  020001  001126             BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
     023454  012737  011700  001124             MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
     023462  013737  001124  001166             MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
     023470  005137  001166                     COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
     023474  013737  001126  001164             MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
     023502  043737  001166  001164             BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
```

```
023510  023737  001124  001164          CMP     $GDDAT,$TMPO     ;ARE THE EXPECTED STATUS BITS SET ?
023516  001401                          BEQ     65$              ;BR IF THEY ARE
023520  104005                          EMT     5
023522  000240                  65$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

023524  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
023532  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
023540  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

023546  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
023552  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
023560  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
023564  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
023572  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
023600  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
023606  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
023614  013737  001170  001164          MOV     $TMP2,$TMPO      ;COPY IT INTO '$TMPO'
023622  042737  100100  001164          BIC     #ATA!VV,$TMPO    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023630  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
023636  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
023644  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
023652  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
023660  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023666  023737  001164  001166          CMP     $TMPO,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023674  001006                          BNE     66$              ;BR IF NOT
023676  005737  001164                  TST     $TMPO            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
023702  001045                          BNE     68$              ;BR IF NOT
023704  104046                          EMT     46
023706  000137  024106                  JMP     70$              ;BYPASS THE REST OF THE CHECKS
023712  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023720  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023726  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
023734  005737  001164                  TST     $TMPO            ;SEE IF STATUS EQ 0 FROM PORT A.
023740  001414                          BEQ     67$              ;BR IF ZERO
023742  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023750  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
023756  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
023764  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
023770  001012                          BNE     68$              ;BR IF NOT
023772  012737  177777  001254  67$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
024000  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
024006  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
024014  104026                          EMT     26
024016  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
024024  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
024032  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
024040  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
024046  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
024050  104007                          EMT     7
024052  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
024060  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
024066  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
024074  023737  001124  001126          CMP     $GDDAT,$RDDAT    ;SEE IF READ OK FROM PORT B.
```

```
024102  001401                          BEQ     70$             ;BR IF OK
024104  104007                          EMT     7
024106  000240                  70$:    NOP
024110  000004                  1$:     SCOPE                   ;LOOP ?
648
660
661                             ;;*****************************************************************
                                ;*TEST 17        SEIZE 'B' BY RMCS1 TEST
                                ;*
                                ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
                                ;*IF THE DRIVE IS IN NEUTRAL.
                                ;*  A.  READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT
                                ;*      THE DRIVE IS SEIZED.
                                ;*
                                ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
024112                          TST17:
024112  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
024116  001406                          BEQ     2$              ;BR IF NOT
024120  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
024122  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
024126  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
024134  012737  024150  001106  2$:     MOV     #TEST17,$LPADR  ;SETUP SCOPE LOOP ADDRESS
024142  012737  024150  001110          MOV     #TEST17,$LPERR  ;SETUP ERROR LOOP ADDRESS
024150                          TEST17:
024150  112737  000017  001102          MOVB    #17,$TSTNM      ;MOVE #17 TO TEST NUMBER
024156  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
024162  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
662
663
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

024170  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
024176  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
024202  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
024210  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
024216  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
024224  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
024230  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
024236  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

024244  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
024252  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
024260  005760  000000                  TST     RMCS1(R0)       ;READ RMCS1
024264  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
024272  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024300  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
024306  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
024314  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
024320  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
024326  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
024332  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
024340  001403                          BEQ     64$             ;BR IF IT IS
```

```
024342  104004                          EMT     4
024344  000137  025034                  JMP     1$              ;BYPASS REST OF THE SUBTEST
024350                          64$:
024350  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
024356  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024364  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
024372  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
024400  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
024406  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
024414  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
024420  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
024426  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
024434  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
024442  001401                          BEQ     65$             ;BR IF THEY ARE
024444  104005                          EMT     5
024446  000240                  65$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

024450  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
024456  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024464  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

024472  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
024476  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
024504  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
024510  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
024516  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
024524  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
024532  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
024540  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
024546  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024554  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
024562  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
024570  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
024576  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
024604  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024612  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
024620  001006                          BNE     66$             ;BR IF NOT
024622  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME  ARE THEY ZERO ?
024626  001045                          BNE     68$             ;BR IF NOT
024630  104046                          EMT     46
024632  000137  025032                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
024636  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
024644  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024652  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
024660  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
024664  001414                          BEQ     67$             ;BR IF ZERO
024666  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024674  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
024702  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
024710  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
024714  001012                          BNE     68$             ;BR IF NOT
024716  012737  177777  001254  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
024724  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
```

```
024732  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
024740  104026                          EMT     26
024742  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
024750  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
024756  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
024764  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
024772  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
024774  104007                          EMT     7
024776  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
025004  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
025012  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
025020  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
025026  001401                          BEQ     70$              ;BR IF OK
025030  104007                          EMT     7
025032  000240                  70$:    NOP
025034  000004                  1$:     SCOPE                    ;LOOP ?
```

```
664
680
681
                                ;;*****************************************************************
                                ;*TEST 20        PORT 'A' INHIBIT SEIZE BY RMCS1 TEST
                                ;*
                                ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
                                ;*      REQUEST' IF THE DRIVE IS SEIZED.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1.  VERIFY THAT
                                ;*      THE DRIVE HAS BEEN SEIZED.
                                ;*
                                ;*  B.  READ THE CONTROL REGISTER FROM PORT 'A'.  VERIFY THAT 'DVA' IS NOT
                                ;*      SET.
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
025036                          TST20:
025036  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
025042  001406                          BEQ     2$               ;BR IF NOT
025044  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
025046  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
025052  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
025060  012737  025074  001106  2$:     MOV     #TEST20,$LPADR   ;SETUP SCOPE LOOP ADDRESS
025066  012737  025074  001110          MOV     #TEST20,$LPERR   ;SETUP ERROR LOOP ADDRESS
025074                          TEST20:
025074  112737  000020  001102          MOVB    #20,$TSTNM       ;MOVE #20 TO TEST NUMBER
025102  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
025106  012737  000012  001176          MOV     #10.,$TIMES      ;;DO 10. ITERATIONS
682
699
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

025114  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT #A
025122  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE
025126  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
025134  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
025142  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT #B
025150  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
025154  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
```

```
025162  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

025170  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
025176  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
025204  005760  000000                  TST     RMCS1(R0)       ;READ RMCS1
025210  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
025216  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025224  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
025232  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
025240  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
025244  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
025252  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
025256  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
025264  001403                          BEQ     64$             ;BR IF IT IS
025266  104004                          EMT     4
025270  000137  026102                  JMP     1$              ;BYPASS REST OF THE SUBTEST
025274                          64$:
025274  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
025302  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025310  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
025316  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
025324  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
025332  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
025340  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
025344  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
025352  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
025360  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
025366  001401                          BEQ     65$             ;BR IF THEY ARE
025370  104005                          EMT     5
025372  000240                  65$:    NOP
025374  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
025402  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

025410  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
025414  016037  000000  001126          MOV     RMCS1(R0),$BDDAT  ;GET CONTENTS OF RMCS.
025422  012737  000000  001122          MOV     #RMCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
025430  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
025434  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
025440  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
025446  042737  173700  001164          BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
025454  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
025462  001414                          BEQ     66$             ;BR IF OK
025464  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
025472  042737  004077  001174          BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
025500  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
025506  104010                          EMT     10
025510  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
025514  000240                  66$:    NOP

                                ;DRIVE SHOULD RETURN TO NEUTRAL

                                ;RELEASE THE DRIVE FROM PORT B
```

```
025516  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
025524  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025532  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

025540  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
025544  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
025552  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
025556  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
025564  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
025572  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
025600  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
025606  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
025614  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025622  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
025630  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
025636  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
025644  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
025652  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025660  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
025666  001006                          BNE     68$             ;BR IF NOT
025670  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
025674  001045                          BNE     70$             ;BR IF NOT
025676  104046                          EMT     46
025700  000137  026100                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
025704  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
025712  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025720  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
025726  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
025732  001414                          BEQ     69$             ;BR IF ZERO
025734  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025742  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
025750  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
025756  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
025762  001012                          BNE     70$             ;BR IF NOT
025764  012737  177777  001254  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
025772  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
026000  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
026006  104026                          EMT     26
026010  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
026016  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
026024  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
026032  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
026040  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
026042  104007                          EMT     7
026044  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
026052  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
026060  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
026066  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
026074  001401                          BEQ     72$             ;BR IF OK
026076  104007                          EMT     7
026100  000240                  72$:    NOP
026102  000004                  1$:     SCOPE                   ;LOOP ?
700
716
717                             ;;*****************************************************
```

```
                                        ;*TEST 21       PORT 'B' INHIBIT SEIZE BY RMCS1 TEST
                                        ;*
                                        ;*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
                                        ;*      REQUEST' IF THE DRIVE IS SEIZED.
                                        ;*
                                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1.  VERIFY THAT
                                        ;*      THE DRIVE HAS BEEN SEIZED.
                                        ;*
                                        ;*  B.  READ THE CONTROL REGISTER FROM PORT 'B'.  VERIFY THAT 'DVA' IS NOT
                                        ;*      SET.
                                        ;*
                                        ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
     026104                             TST21:
     026104  005737  001300                     TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
     026110  001406                             BEQ     2$              ;BR IF NOT
     026112  100002                             BPL     1$              ;BR IF JUST ENTERED TEST
     026114  000137  003062                     JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
     026120  012737  177777  001300     1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
     026126  012737  026142  001106     2$:     MOV     #TEST21,$LPADR  ;SETUP SCOPE LOOP ADDRESS
     026134  012737  026142  001110             MOV     #TEST21,$LPERR  ;SETUP ERROR LOOP ADDRESS
     026142                             TEST21:
     026142  112737  000021  001102             MOVB    #21,$TSTNM      ;MOVE #21 TO TEST NUMBER
     026150  012706  001100                     MOV     #STACK,SP       ;LOAD THE STACK POINTER
     026154  012737  000012  001176             MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
718
719
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

     026162  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
     026170  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE
     026174  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     026202  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
     026210  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
     026216  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
     026222  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     026230  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

     026236  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
     026244  013737  001224  001242             MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
     026252  005760  000000                     TST     RMCS1(R0)       ;READ RMCS1
     026256  113760  001226  000010             MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
     026264  013737  001226  001240             MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     026272  013737  001226  001244             MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
     026300  016037  000012  001126             MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
     026306  010037  001122                     MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
     026312  062737  000012  001122             ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
     026320  005037  001124                     CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
     026324  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
     026332  001403                             BEQ     64$             ;BR IF IT IS
     026334  104004                             EMT     4
     026336  000137  027150                     JMP     1$              ;BYPASS REST OF THE SUBTEST
     026342                             64$:
```

```
026342  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
026350  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026356  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
026364  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
026372  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
026400  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
026406  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
026412  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
026420  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
026426  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
026434  001401                          BEQ     65$             ;BR IF THEY ARE
026436  104005                          EMT     5
026440  000240                  65$:    NOP
026442  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
026450  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRLSS TO LOCATION FOR TYPEOUT

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

026456  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
026462  016037  000000  001126          MOV     RMCS1(R0),$BDDAT  ;GET CONTENTS OF RMCS1
026470  012737  000000  001122          MOV     #RMCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
026476  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
026502  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
026506  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
026514  042737  173700  001164          BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
026522  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
026530  001414                          BEQ     66$             ;BR IF OK
026532  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
026540  042737  004077  001174          BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
026546  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GUOD DATA FOR TYPEOUT
026554  104010                          EMT     10
026556  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
026562  000240                  66$:    NOP

                                ;DRIVE SHCULD RETURN TO NEUTRAL

                                ;RELEASE THE DRIVE FROM PORT A

026564  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
026572  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026600  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

026606  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
026612  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
026620  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
026624  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
026632  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
026640  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
026646  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
026654  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
026662  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026670  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
026676  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
026704  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
026712  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
```

```
026720  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026726  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
026734  001006                          BNE     68$              ;BR IF NOT
026736  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
026742  001045                          BNE     70$              ;BR IF NOT
026744  104046                          EMT     46
026746  000137  027146                  JMP     72$              ;BYPASS THE REST OF THE CHECKS
026752  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
026760  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
026766  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
026774  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
027000  001414                          BEQ     69$              ;BR IF ZERO
027002  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027010  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
027016  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
027024  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
027030  001012                          BNE     70$              ;BR IF NOT
027032  012737  177777  001254  69$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
027040  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
027046  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
027054  104026                          EMT     26
027056  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
027064  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
027072  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
027100  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
027106  001401                          BEQ     71$              ;BR IF OK FROM PORT A.
027110  104007                          EMT     7
027112  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
027120  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
027126  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
027134  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
027142  001401                          BEQ     72$              ;BR IF OK
027144  104007                          EMT     7
027146  000240                  72$:    NOP
027150  000004                  1$:     SCOPE                    ;LOOP ?
720
735
736                             ;;*****************************************************************
                                ;*TEST 22        SEIZE BY RMAS TEST
                                ;*
                                ;*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
                                ;*      (RMAS) SEIZES THE DRIVE.  VERIFY THAT REQUEST IS SET FOR THE OTHER
                                ;*      PORT.
                                ;*
                                ;*  A.  WRITE THE APPROPRIATE DRIVE BIT INTO RMAS; VERIFY THAT THE DRIVE
                                ;*      IS SEIZED.
                                ;*
                                ;*  B.  ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
                                ;*      DRIVE SWITCHES TO THE OPPOSITE PORT.  ISSUE A RELEASE THROUGH THE
                                ;*      OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
                                ;*
                                ;;*************** ***************************************************
027152                          TST22:
027152  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
027156  001406                          BEQ     2$               ;BR IF NOT
027160  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
027162  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
```

```
027166  012737  177777  001300  1$:     MOV     #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
027174  012737  027210  001106  2$:     MOV     #TEST22,$LPADR    ;SETUP SCOPE LOOP ADDRESS
027202  012737  027210  001110          MOV     #TEST22,$LPERR    ;SETUP ERROR LOOP ADDRESS
027210                          TEST22:
027210  112737  000022  001102          MOVB    #22,$TSTNM        ;MOVE #22 TO TEST NUMBER
027216  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
027222  012737  000012  001176          MOV     #10.,$TIMES       ;;DO 10. ITERATIONS
737
791
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

027230  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
027236  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
027242  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
027250  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
027256  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
027264  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
027270  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
027276  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                ;SELECT DRIVE OTHER THAN THAT BEING TESTED

027304  113760  001230  000010          MOVB    PORTC,RMCS2(R0) ;SELECT DRIVE NOT BEING TESTED
027312  013737  001224  001242          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS

                                ;WRITE THE DRIVE'S ATTENTION BIT

027320  013760  001236  000016          MOV     ASR1,RMAS(R0)   ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
027326  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
027334  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

027342  005760  000012                  TST     RMDS(R0)        ;SEE THE REGISTER THROUGH PORT A ?
027346  001014                          BNE     1$              ;BR IF YES
027350  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
027356  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027364  005760  000012                  TST     RMDS(R0)        ;SEE REGISTER THROUGH PORT B ?
027370  001021                          BNE     2$              ;BR IF YES
027372  104042                          EMT     42
027374  000137  031224                  JMP     4$              ;BYPASS REST OF TEST
027400                          1$:
027400  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
027406  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027414  005760  000012                  TST     RMDS(R0)        ;REGISTER SHOULD BE ZERO THROUGH PORT B
027420  001002                          BNE     .+6             ;BR IF STATUS REG IS NOT ZERO
027422  000137  030324                  JMP     3$              ;STATUS REG IS ZERO
027426  104043                          EMT     43
027430  000137  031224                  JMP     4$              ;BYPASS REST OF TEST

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

027434                          2$:
027434  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
027440  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
027446  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
027454  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
```

```
027460  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
027466  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
027474  042737  106077  001164          BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
027502  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
027510  001414                          BEQ     64$             ;BR IF OK
027512  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
027520  042737  071700  001174          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
027526  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
027534  104010                          EMT     10
027536  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
027542  000240                  64$:    NOP
027544  013737  001226  001242          MOV     PORTB,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
027552  013737  001224  001244          MOV     PORTA,OPPRT     ;SAME AS ABOVE

                                ;RELEASE THE DRIVE FROM PORT B

027560  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
027566  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027574  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

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

027602  005037  001254                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
027606  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
027614  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
027622  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
027626  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
027634  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027642  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT A
027650  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
027656  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027664  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
027672  001404                          BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
027674  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
027700  001401                          BEQ     66$             ;BR IF ZERO
027702  104044                          EMT     44
027704  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
027712  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
027720  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
027726  001401                          BEQ     67$             ;BR IF OK
027730  104027                          EMT     27
027732  000240                  67$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

027734  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
027742  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027750  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

027756  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
027762  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
027770  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
027774  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
030002  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
030010  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
```

```
030016  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
030024  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
030032  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030040  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
030046  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
030054  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
030062  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
030070  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030076  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
030104  001006                          BNE     68$              ;BR IF NOT
030106  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
030112  001045                          BNE     70$              ;BR IF NOT
030114  104046                          EMT     46
030116  000137  030316                  JMP     72$              ;BYPASS THE REST OF THE CHECKS
030122  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
030130  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030136  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
030144  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
030150  001414                          BEQ     69$              ;BR IF ZERO
030152  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030160  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
030166  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
030174  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
030200  001012                          BNE     70$              ;BR IF NOT
030202  012737  177777  001254  69$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
030210  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
030216  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
030224  104026                          EMT     26
030226  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
030234  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
030242  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
030250  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
030256  001401                          BEQ     71$              ;BR IF OK FROM PORT A.
030260  104007                          EMT     7
030262  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
030270  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
030276  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
030304  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
030312  001401                          BEQ     72$              ;BR IF OK
030314  104007                          EMT     7
030316  000240                  72$:    NOP
030320  000137  031224                  JMP     4$

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

030324                          3$:
030324  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
030332  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030340  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
030344  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
030352  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
030360  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
030364  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
030372  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
030400  042737  106077  001164          BIC     #^C71700,$TMP0   ;SAVE SPECIFIED BITS
030406  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
030414  001414                          BEQ     73$              ;BR IF OK
```

```
030416  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
030424  042737  071700  001174          BIC     #71700,$TMP4     ;CLEAR THE MASKED BITS
030432  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
030440  104010                          EMT     10
030442  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
030446  000240                  73$:    NOP
030450  013737  001224  001242          MOV     PORTA,SEIZPT     ;ADDRESS FOR ERROR MESSAGE
030456  013737  001226  001244          MOV     PORTB,OPPRT      ;SAME AS ABOVE

                                ;RELEASE THE DRIVE FROM PORT A

030464  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
030472  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030500  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

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

030506  005037  001254                  CLR     RELERR           ;CLEAR 'RELEASE ERROR' INDICATOR
030512  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
030520  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
030526  060037  001122                  ADD     R0,$BDADR        ;REGISTER BASE ADDRESS FOR TYPEOUT
030532  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
030540  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030546  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT B
030554  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
030562  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030570  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
030576  001404                          BEQ     75$              ;BR IF STATUS FROM PORT A ZERO
030600  005737  001164                  TST     $TMP0            ;IS STATUS FROM PORT B ZERO ?
030604  001401                          BEQ     75$              ;BR IF ZERO
030606  104044                          EMT     44
030610  013737  001164  001126  75$:    MOV     $TMP0,$BDDAT     ;CHECK STATUS FROM PORT B
030616  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT ADDRESS FOR TYPEOUT
030624  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;COMPARE WITH CONSTANT
030632  001401                          BEQ     76$              ;BR IF OK
030634  104027                          EMT     27
030636  000240                  76$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

030640  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
030646  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030654  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

030662  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
030666  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
030674  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
030700  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
030706  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
030714  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
030722  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
030730  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
030736  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030744  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
030752  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
030760  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
030766  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
030774  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031002  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031010  001006                          BNE     77$             ;BR IF NOT
031012  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
031016  001045                          BNE     79$             ;BR IF NOT
031020  104046                          EMT     46
031022  000137  031222                  JMP     81$             ;BYPASS THE REST OF THE CHECKS
031026  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031034  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031042  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
031050  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
031054  001414                          BEQ     78$             ;BR IF ZERO
031056  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031064  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
031072  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
031100  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
031104  001012                          BNE     79$             ;BR IF NOT
031106  012737  177777  001254  78$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
031114  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
031122  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
031130  104026                          EMT     26
031132  013737  001170  001126  79$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
031140  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
031146  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031154  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
031162  001401                          BEQ     80$             ;BR IF OK FROM PORT A.
031164  104007                          EMT     7
031166  013737  001172  001126  80$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
031174  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
031202  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031210  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
031216  001401                          BEQ     81$             ;BR IF OK
031220  104007                          EMT     7
031222  000240                  81$:    NOP
031224  000004                  4$:     SCOPE                   ;LOOP ?
```

792
804
805

```
                                ;;*****************************************************************
                                ;*TEST 23        INHIBIT SEIZE BY RMAS TEST
                                ;*
                                ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
                                ;*      THE DRIVE'S ATTENTION BIT.
                                ;*
                                ;*  A.  SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
                                ;*      BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
                                ;*
                                ;*  B.  VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
                                ;*
                                ;;*****************************************************************
031226                          TST23:
031226  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
031232  001406                          BEQ     2$              ;BR IF NOT
031234  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
031236  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
031242  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
```

```
031250  012737  031264  001106  2$:     MOV     #TEST23,$LPADR  ;SETUP SCOPE LOOP ADDRESS
031256  012737  031264  001110          MOV     #TEST23,$LPERR  ;SETUP ERROR LOOP ADDRESS
031264                          TEST23:
031264  112737  000023  001102          MOVB    #23,$TSTNM      ;MOVE #23 TO TEST NUMBER
031272  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
031276  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
806
821
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

031304  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
031312  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
031316  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
031324  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
031332  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
031340  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
031344  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
031352  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
031360  113760  001230  000010          MOVB    PORTC,RMCS2(R0) ;SELECT DRIVE NOT BEING TESTED

                                ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST

031366  013737  001236  001164          MOV     ASR1,$TMP0      ;STORE ATTN BIT FOR PORT A
031374  005137  001164                  COM     $TMP0           ;COMPLEMENT IT
031400  013760  001164  000016          MOV     $TMP0,RMAS(R0)  ;WRITE THE ATTN REGISTER

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

031406  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
031412  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
031420  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
031424  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
031432  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
031440  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
031446  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
031454  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
031462  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031470  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
031476  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
031504  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
031512  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
031520  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031526  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031534  001006                          BNE     64$             ;BR IF NOT
031536  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
031542  001045                          BNE     66$             ;BR IF NOT
031544  104046                          EMT     46
031546  000137  031746                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
031552  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031560  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031566  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
031574  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
031600  001414                          BEQ     65$             ;BR IF ZERO
031602  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031610  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
031616  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
031624  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
```

```
031630  001012                          BNE     66$             ;BR IF NOT
031632  012737  177777  001254  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
031640  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
031646  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
031654  104021                          EMT     21
031656  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
031664  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
031672  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
031706  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
031710  104007                          EMT     7
031712  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
031720  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
031726  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031734  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
031742  001401                          BEQ     68$             ;BR IF OK
031744  104007                          EMT     7
031746  000240                  68$:    NOP
031750  000004                          SCOPE                   ;LOOP ?
822
841
842
                                ;;*****************************************************************
                                ;*TEST 24        SET PORT 'A' REQUEST TEST
                                ;*
                                ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
                                ;*      DRIVE IS SEIZED BY THE OTHER PORT.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WRITE 0'S INTO RMDS FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
                                ;*      SEIZED BY PORT 'B'.
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
                                ;*      SWITCHED TO PORT 'A'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
                                ;*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
                                ;*
                                ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
031752                          TST24:
031752  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
031756  001406                          BEQ     2$              ;BR IF NOT
031760  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
031762  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
031766  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
031774  012737  032010  001106  2$:     MOV     #TEST24,$LPADR  ;SETUP SCOPE LOOP ADDRESS
032002  012737  032010  001110          MOV     #TEST24,$LPERR  ;SETUP ERROR LOOP ADDRESS
032010                          TEST24:
032010  112737  000024  001102          MOVB    #24,$TSTNM      ;MOVE #24 TO TEST NUMBER
032016  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
032022  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
843
872
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

032030  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
```

```
032036  005060  000012                  CLR     RMDS(R0)          ;SEIZE THE DRIVE
032042  012760  000011  000000          MOV     #11,RMCS1(R0)     ;ISSUE DRIVE CLEAR
032050  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
032056  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT #B
032064  005060  000012                  CLR     RMDS(R0)          ;SEIZE THE DRIVE THROUGH PORT 'B'
032070  012760  000011  000000          MOV     #11,RMCS1(R0)     ;ISSUE DRIVE CLEAR
032076  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

032104  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
032112  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
032120  005060  000012                  CLR     RMDS(R0)          ;WRITE RMDS
032124  013737  001224  001244          MOV     PORTA,OPPRT       ;'OPPOSITE' PORT ADDRESS
032132  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
032140  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;SET PORT REQUEST

032146  005060  000012                  CLR     RMDS(R0)          ;SET PORT REQUEST FOR PORT A

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

                                ;RELEASE THE DRIVE FROM PORT B

032152  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
032160  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032166  012760  000013  000000          MOV     #13,RMCS1(R0)     ;ISSUE RELEASE THROUGH PORT B

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

032174  005037  001254                  CLR     RELERR            ;CLEAR 'RELEASE ERROR' INDICATOR
032200  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
032206  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
032214  060037  001122                  ADD     R0,$BDADR         ;REGISTER BASE ADDRESS FOR TYPEOUT
032220  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
032226  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032234  016037  000012  001164          MOV     RMDS(R0),$TMP0    ;READ STATUS REGISTER FROM PORT A
032242  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
032250  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032256  016037  000012  001126          MOV     RMDS(R0),$BDDAT   ;DRIVE STATUS FROM PORT B
032264  001404                          BEQ     66$               ;BR IF STATUS FROM PORT B ZERO
032266  005737  001164                  TST     $TMP0             ;IS STATUS FROM PORT A ZERO ?
032272  001401                          BEQ     66$               ;BR IF ZERO
032274  104031                          EMT     31
032276  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT      ;CHECK STATUS FROM PORT A
032304  013737  001224  001240          MOV     PORTA,PTNBR       ;CHANGE PORT ADDRESS FOR TYPEOUT
032312  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;COMPARE WITH CONSTANT
032320  001401                          BEQ     67$               ;BR IF OK
032322  104027                          EMT     27
032324  000240                  67$:    NOP
032326  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
032334  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032342  005037  001250                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
032346  016037  000012  001126          MOV     RMDS(R0),$BDDAT   ;GET CONTENTS OF RMDS
032354  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032362  060037  001122                  ADD     R0,$BDADR         ;ADD RH/RM BASE ADDRESS
```

```
032366  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
032372  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
032400  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
032406  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
032414  001414                          BEQ     68$             ;BR IF OK
032416  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
032424  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
032432  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
032440  104016                          EMT     16
032442  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
032446  000240                  68$:    NOP
032450  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
032456  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032464  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
032470  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
032476  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032504  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
032510  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
032516  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
032524  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
032532  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
032540  001414                          BEQ     70$             ;BR IF OK
032542  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
032550  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
032556  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
032564  104016                          EMT     16
032566  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
032572  000240                  70$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

032574  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
032602  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032610  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

032616  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
032622  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
032630  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
032634  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
032642  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
032650  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
032656  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
032664  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
032672  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032700  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
032706  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
032714  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
032722  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
032730  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032736  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
032744  001006                          BNE     72$             ;BR IF NOT
032746  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
032752  001045                          BNE     74$             ;BR IF NOT
032754  104046                          EMT     46
032756  000137  033142                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
```

```
032762  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
032770  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
032776  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
033004  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
033010  001414                          BEQ     73$              ;BR IF ZERO
033012  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033020  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
033026  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
033034  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
033040  001012                          BNE     74$              ;BR IF NOT
033042  012737  177777  001254  73$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
033050  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
033056  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
033064  104026                          EMT     26
033066  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
033074  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
033102  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
033110  001401                          BEQ     75$              ;BR IF OK FROM PORT A.
033112  104007                          EMT     7
033114  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
033122  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
033130  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
033136  001401                          BEQ     76$              ;BR IF OK
033140  104007                          EMT     7
033142  000240                  76$:    NOP
033144  000004                  1$:     SCOPE                    ;LOOP ?
```

```
;;*****************************************************************
;;*TEST 25        SET PORT 'B' REQUEST TEST
;;*
;;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
;;*      DRIVE IS SEIZED BY THE OTHER PORT.
;;*
;;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
;;*
;;*  B.  WRITE 0'S INTO RMDS 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.
;;*
;;*****************************************************************
```

```
033146          TST25:
033146  005737  001300          TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
033152  001406                  BEQ     2$               ;BR IF NOT
033154  100002                  BPL     1$               ;BR IF JUST ENTERED TEST
033156  000137  003062          JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
033162  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
033170  012737  033204  001106  2$:     MOV     #TEST25,$LPADR   ;SETUP SCOPE LOOP ADDRESS
033176  012737  033204  001110          MOV     #TEST25,$LPERR   ;SETUP ERROR LOOP ADDRESS
033204          TEST25:
033204  112737  000025  001102          MOVB    #25,$TSTNM       ;MOVE #25 TO TEST NUMBER
```

```
033212  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
033216  012737  000012  001176          MOV     #10.,$TIMES       ;;DO 10. ITERATIONS

                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

033224  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
033232  005060  000012                  CLR     RMDS(R0)          ;SEIZE THE DRIVE
033236  012760  000011  000000          MOV     #11,RMCS1(R0)     ;ISSUE DRIVE CLEAR
033244  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
033252  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
033260  005060  000012                  CLR     RMDS(R0)          ;SEIZE THE DRIVE THROUGH PORT 'B'
033264  012760  000011  000000          MOV     #11,RMCS1(R0)     ;ISSUE DRIVE CLEAR
033272  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT A

033300  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
033306  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
033314  005060  000012                  CLR     RMDS(R0)          ;WRITE RMDS
033320  013737  001226  001244          MOV     PORTB,OPPRT       ;'OPPOSITE' PORT ADDRESS
033326  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
033334  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;SET PORT REQUEST

033342  005060  000012                  CLR     RMDS(R0)          ;SET PORT REQUEST FOR PORT B

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

                                ;RELEASE THE DRIVE FROM PORT A

033346  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
033354  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033362  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

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

033370  005037  001254                  CLR     RELERR            ;CLEAR 'RELEASE ERROR' INDICATOR
033374  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
033402  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
033410  060037  001122                  ADD     R0,$BDADR         ;REGISTER BASE ADDRESS FOR TYPEOUT
033414  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
033422  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033430  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT B
033436  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
033444  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033452  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
033460  001404                          BEQ     66$               ;BR IF STATUS FROM PORT A ZERO
033462  005737  001164                  TST     $TMP0             ;IS STATUS FROM PORT B ZERO ?
033466  001401                          BEQ     66$               ;BR IF ZERO
033470  104031                          EMT     31
033472  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT      ;CHECK STATUS FROM PORT B
033500  013737  001226  001240          MOV     PORTB,PTNBR       ;CHANGE PORT ADDRESS FOR TYPEOUT
033506  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;COMPARE WITH CONSTANT
033514  001401                          BEQ     67$               ;BR IF OK
033516  104027                          EMT     27
```

```
033520  000240                  67$:    NOP
033522  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
033530  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033536  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
033542  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
033550  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033556  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
033562  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
033566  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
033574  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
033602  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
033610  001414                          BEQ     68$              ;BR IF OK
033612  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
033620  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
033626  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
033634  104016                          EMT     16
033636  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
033642  000240                  68$:    NOP
033644  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
033652  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033660  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
033664  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
033672  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033700  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
033704  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
033712  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
033720  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
033726  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
033734  001414                          BEQ     70$              ;BR IF OK
033736  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
033744  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
033752  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
033760  104016                          EMT     16
033762  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
033766  000240                  70$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

033770  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
033776  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034004  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

034012  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
034016  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
034024  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
034030  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
034036  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
034044  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
034052  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
034060  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
034066  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034074  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
034102  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
034110  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
034116  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
```

```
034124  042737  100100  001166          BIC     #AIA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034132  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
034140  001006                          BNE     72$              ;BR IF NOT
034142  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
034146  001045                          BNE     74$              ;BR IF NOT
034150  104046                          EMT     46
034152  000137  034336                  JMP     76$              ;BYPASS THE REST OF THE CHECKS
034156  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
034164  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034172  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
034200  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
034204  001414                          BEQ     73$              ;BR IF ZERO
034206  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034214  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
034222  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
034230  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
034234  001012                          BNE     74$              ;BR IF NOT
034236  012737  177777  001254  73$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
034244  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
034252  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
034260  104026                          EMT     26
034262  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
034270  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
034276  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
034304  001401                          BEQ     75$              ;BR IF OK FROM PORT A.
034306  104007                          EMT     7
034310  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
034316  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
034324  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
034332  001401                          BEQ     76$              ;BR IF OK
034334  104007                          EMT     7
034336  000240                  76$:    NOP
034340  000004                  1$:     SCOPE                    ;LOOP ?
899
918
919
                                ;;*****************************************************************
                                ;;*TEST 26        TEST RESET ATTENTION 'A' BY DRIVE CLEAR
                                ;;*
                                ;;*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
                                ;;*      SEIZING PORT.
                                ;;*
                                ;;*  A.  SET EACH PORT 'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
                                ;;*      SET.
                                ;;*
                                ;;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;;*
                                ;;*  C.  ISSUE A DRIVE CLEAR COMMAND.
                                ;;*
                                ;;*
                                ;;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE ATTENTION
                                ;;*      BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
                                ;;*      'B' IS STILL SET.
                                ;;*
                                ;;*****************************************************************
034342                          TST26:
034342  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
034346  001406                          BEQ     2$               ;BR IF NOT
```

```
034350  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
034352  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
034356  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
034364  012737  034400  001106  2$:     MOV     #TEST26,$LPADR  ;SETUP SCOPE LOOP ADDRESS
034372  012737  034400  001110          MOV     #TEST26,$LPERR  ;SETUP ERROR LOOP ADDRESS
034400                          TEST26:
034400  112737  000026  001102          MOVB    #26,$TSTNM      ;MOVE #26 TO TEST NUMBER
034406  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
034412  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
920
953
                                ;SET ATTENTION BITS FOR BOTH PORTS

034420  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
034426  005760  000012          66$:    TST     RMDS(R0)        ;MAKE SURE DRIVE  AVAILABLE
034432  001775                          BEQ     66$
034434  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS
034442  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
034446  013760  001226  000010          MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
034454  005760  000012          64$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
034460  001775                          BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
034462  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS ON PORT 65$
034470  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
034474  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT ''64$'' AGAIN
034502  005760  000012          65$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
034506  001775                          BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT

                                ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

034510  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
034516  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034524  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
034530  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
034536  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034544  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
034550  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
034556  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
034564  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
034572  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
034600  001414                          BEQ     67$             ;BR IF OK
034602  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
034610  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
034616  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
034624  104010                          EMT     10
034626  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
034632  000240                  67$:    NOP
034634  005737  001250                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
034640  001402                          BEQ     .+6             ;BR IF IT WAS
034642  000137  036034                  JMP     1$              ;BYPASS REST OF TEST IF NOT
034646  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
034654  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034662  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
034666  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
034674  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034702  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
034706  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
034714  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
```

```
034722  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
034730  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
034736  001414                          BEQ     69$             ;BR IF OK
034740  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
034746  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
034754  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
034762  104010                          EMT     10
034764  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
034770  000240                  69$:    NOP
034772  005737  001250                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
034776  001402                          BEQ     .+6             ;BR IF IT WAS
035000  000137  036034                  JMP     1$              ;BYPASS REST OF TEST IF NOT

                                ;SEIZE THE DRIVE THROUGH PORT A

035004  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
035012  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
035020  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
035024  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
035032  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035040  013737  001226  001244          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
035046  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
035054  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
035060  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
035066  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
035072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
035100  001403                          BEQ     71$             ;BR IF IT IS
035102  104004                          EMT     4
035104  000137  036034                  JMP     1$              ;BYPASS REST OF THE SUBTEST
035110                          71$:
035110  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
035116  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035124  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
035132  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
035140  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
035146  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
035154  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
035160  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
035166  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
035174  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
035202  001401                          BEQ     72$             ;BR IF THEY ARE
035204  104005                          EMT     5
035206  000240                  72$:    NOP

                                ;ISSUE DRIVE CLEAR COMMAND TO PORT A

035210  012760  000011  000000          MOV     #11,RMCS1(R0)   ;DO A DRIVE CLEAR COMMAND

                                ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED

035216  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
035222  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
035230  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
035236  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
035242  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
035246  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
035254  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
```

```
035262  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
035270  001414                          BEQ     73$              ;BR IF OK
035272  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
035300  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
035306  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
035314  104047                          EMT     47
035316  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
035322  000240                  73$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

035324  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
035332  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035340  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

035346  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
035352  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
035360  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
035364  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
035372  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
035400  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
035406  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
035414  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
035422  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035430  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
035436  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
035444  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
035452  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
035460  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035466  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
035474  001006                          BNE     75$              ;BR IF NOT
035476  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
035502  001045                          BNE     77$              ;BR IF NOT
035504  104046                          EMT     46
035506  000137  035706                  JMP     79$              ;BYPASS THE REST OF THE CHECKS
035512  013737  001170  001126  75$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
035520  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035526  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
035534  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
035540  001414                          BEQ     76$              ;BR IF ZERO
035542  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035550  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
035556  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
035564  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
035570  001012                          BNE     77$              ;BR IF NOT
035572  012737  177777  001254  76$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
035600  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
035606  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
035614  104026                          EMT     26
035616  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
035624  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
035632  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
035640  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
035646  001401                          BEQ     78$              ;BR IF OK FROM PORT A.
035650  104007                          EMT     7
```

```
035652  013737  001172  001126  78$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
035660  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
035666  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
035674  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
035702  001401                          BEQ     79$              ;BR IF OK
035704  104007                          EMT     7
035706  000240                  79$:    NOP

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

035710  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
035716  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035724  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
035730  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
035736  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
035744  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
035750  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
035756  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
035764  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
035772  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
036000  001414                          BEQ     80$              ;BR IF OK
036002  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
036010  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
036016  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
036024  104050                          EMT     50
036026  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
036032  000240                  80$:    NOP
036034  000004                  1$:     SCOPE                    ;LOOP ?
954
972
973
                                ;*****************************************************************
                                ;*TEST 27        TEST RESET ATTENTION 'B' BY DRIVE CLEAR
                                ;*
                                ;*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
                                ;*      SEIZING PORT.
                                ;*
                                ;*  A.  SET EACH PORT'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
                                ;*      SET.
                                ;*
                                ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  C.  ISSUE A DRIVE CLEAR COMMAND.
                                ;*
                                ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE ATTENTION
                                ;*      BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
                                ;*      'A' IS STILL SET.
                                ;*
                                ;*****************************************************************
036036                          TST27:
036036  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
036042  001406                          BEQ     2$               ;BR IF NOT
036044  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
036046  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
036052  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
036060  012737  036074  001106  2$:     MOV     #TEST27,$LPADR   ;SETUP SCOPE LOOP ADDRESS
036066  012737  036074  001110          MOV     #TEST27,$LPERR   ;SETUP ERROR LOOP ADDRESS
036074                          TEST27:
```

```
036074  112737  000027  001102          MOVB    #27,$TSTNM      ;MOVE #27 TO TEST NUMBER
036102  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
036106  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

                                ;SET ATTENTION BITS FOR BOTH PORTS

036114  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
036122  005760  000012          66$:    TST     RMDS(R0)        ;MAKE SURE DRIVE  AVAILABLE
036126  001775                          BEQ     66$
036130  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS
036136  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
036142  013760  001226  000010          MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
036150  005760  000012          64$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
036154  001775                          BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
036156  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS ON PORT 65$
036164  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
036170  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT "64$" AGAIN
036176  005760  000012          65$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
036202  001775                          BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT

                                ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

036204  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
036212  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036220  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
036224  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
036232  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036240  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
036244  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
036252  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
036260  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
036266  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
036274  001414                          BEQ     67$             ;BR IF OK
036276  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
036304  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
036312  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
036320  104010                          EMT     10
036322  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
036326  000240                  67$:    NOP
036330  005737  001250                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
036334  001402                          BEQ     .+6             ;BR IF IT WAS
036336  000137  037530                  JMP     1$              ;BYPASS REST OF TEST IF NOT
036342  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
036350  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036356  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
036362  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
036370  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036376  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
036402  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
036410  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
036416  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
036424  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
036432  001414                          BEQ     69$             ;BR IF OK
036434  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
036442  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
036450  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
036456  104010                          EMT     10
036460  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
036464  000240                  69$:    NOP
036466  005737  001250                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
036472  001402                          BEQ     .+6             ;BR IF IT WAS
036474  000137  037530                  JMP     1$              ;BYPASS REST OF TEST IF NOT

                                ;SEIZE THE DRIVE THROUGH PORT B

036500  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
036506  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
036514  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
036520  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
036526  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036534  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
036542  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
036550  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
036554  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
036562  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
036566  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
036574  001403                          BEQ     71$             ;BR IF IT IS
036576  104004                          EMT     4
036600  000137  037530                  JMP     1$              ;BYPASS REST OF THE SUBTEST
036604                          71$:
036604  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
036612  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036620  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
036626  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
036634  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
036642  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
036650  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
036654  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
036662  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
036670  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
036676  001401                          BEQ     72$             ;BR IF THEY ARE
036700  104005                          EMT     5
036702  000240                  72$:    NOP

                                ;ISSUE DRIVE CLEAR COMMAND TO PORT B

036704  012760  000011  000000          MOV     #11,RMCS1(R0)   ;DO A DRIVE CLEAR COMMAND

                                ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED

036712  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
036716  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
036724  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036732  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
036736  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
036742  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
036750  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
036756  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
036764  001414                          BEQ     73$             ;BR IF OK
036766  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
036774  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
037002  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
037010  104047                          EMT     47
```

```
037012  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
037016  000240                  73$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

037020  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
037026  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037034  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

037042  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
037046  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
037054  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
037060  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
037066  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
037074  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
037102  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
037110  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
037116  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037124  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
037132  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
037140  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
037146  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
037154  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037162  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
037170  001006                          BNE     75$             ;BR IF NOT
037172  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
037176  001045                          BNE     77$             ;BR IF NOT
037200  104046                          EMT     46
037202  000137  037402                  JMP     79$             ;BYPASS THE REST OF THE CHECKS
037206  013737  001170  001126  75$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
037214  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037222  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
037230  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
037234  001414                          BEQ     76$             ;BR IF ZERO
037236  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037244  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
037252  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
037260  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
037264  001012                          BNE     77$             ;BR IF NOT
037266  012737  177777  001254  76$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
037274  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
037302  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
037310  104026                          EMT     26
037312  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
037320  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
037326  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
037334  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
037342  001401                          BEQ     78$             ;BR IF OK FROM PORT A.
037344  104007                          EMT     7
037346  013737  001172  001126  78$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
037354  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
037362  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
037370  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
037376  001401                          BEQ     79$             ;BR IF OK
037400  104007                          EMT     7
```

```
037402  000240                    79$:    NOP

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

037404  113760  001224  000010            MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
037412  013737  001224  001240            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037420  005037  001250                    CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
037424  016037  000012  001126            MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
037432  012737  000012  001122            MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037440  060037  001122                    ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
037444  012737  100000  001124            MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
037452  013737  001126  001164            MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
037460  042737  077777  001164            BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
037466  023737  001124  001164            CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
037474  001414                            BEQ     80$              ;BR IF OK
037476  013737  001126  001174            MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
037504  042737  100000  001174            BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
037512  053737  001174  001124            BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
037520  104050                            EMT     50
037522  005137  001250                    COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
037526  000240                    80$:    NOP
037530  000004                    1$:     SCOPE                    ;LOOP ?
976
995
996
                                  ;;*****************************************************************
                                  ;*TEST 30       RESET ATTENTION 'A' BY GO TEST
                                  ;*
                                  ;* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
                                  ;*      SEIZING PORT.
                                  ;*
                                  ;* A.   SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
                                  ;*      ATTENTION BITS ARE SET.
                                  ;*
                                  ;* B.   SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S
                                  ;*      INTO RMDS.
                                  ;*
                                  ;* C.   ISSUE A NOP COMMAND.
                                  ;*
                                  ;* D.   RELEASE THE FRIVE THROUGH PORT 'A'. VERIFY THAT THE
                                  ;*      ATTENTION BIT FOR PORT 'A' IS RESET, AND THE
                                  ;*      ATTENTION BIT FOR PORT 'B' IS STIL SET.
                                  ;*
                                  ;;*****************************************  ********************
037532                            TST30:
037532  005737  001300                    TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
037536  001406                            BEQ     2$               ;BR IF NOT
037540  100002                            BPL     1$               ;BR IF JUST ENTERED TEST
037542  000137  003062                    JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
037546  012737  177777  001300    1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
037554  012737  037570  001106    2$:     MOV     #TEST30,$LPADR   ;SETUP SCOPE LOOP ADDRESS
037562  012737  037570  001110            MOV     #TEST30,$LPERR   ;SETUP ERROR LOOP ADDRESS
037570                            TEST30:
037570  112737  000030  001102            MOVB    #30,$TSTNM       ;MOVE #30 TO TEST NUMBER
037576  012706  001100                    MOV     #STACK,SP        ;LOAD THE STACK POINTER
037602  012737  000012  001176            MOV     #10.,$TIMES      ;;DO 10. ITERATIONS
997
1030
```

```
                                        ;SET ATTENTION BITS FOR BOTH PORTS

037610  113760  001224  000010           MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
037616  005760  000012          66$:     TST     RMDS(R0)        ;MAKE SURE DRIVE  AVAILABLE
037622  001775                           BEQ     66$
037624  012760  177777  000014           MOV     #-1,RMER1(R0)   ;FORCE ERRORS
037632  005060  000014                   CLR     RMER1(R0)       ;CLEAR THE ERRORS
037636  013760  001226  000010           MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
037644  005760  000012          64$:     TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
037650  001775                           BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
037652  012760  177777  000014           MOV     #-1,RMER1(R0)   ;FORCE ERRORS ON PORT 65$
037660  005060  000014                   CLR     RMER1(R0)       ;CLEAR THE ERRORS
037664  113760  001224  000010           MOVB    PORTA,RMCS2(R0) ;SELECT PORT ''64$'' AGAIN
037672  005760  000012          65$:     TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
037676  001775                           BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT

                                        ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

037700  113760  001224  000010           MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
037706  013737  001224  001240           MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037714  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
037720  016037  000012  001126           MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
037726  012737  000012  001122           MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037734  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
037740  012737  100000  001124           MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
037746  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
037754  042737  077777  001164           BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
037762  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
037770  001414                           BEQ     67$             ;BR IF OK
037772  013737  001126  001174           MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
040000  042737  100000  001174           BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
040006  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
040014  104010                           EMT     10
040016  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
040022  000240                  67$:     NOP
040024  005737  001250                   TST     CKERR           ;WAS ATTENTION SET FOR A??
040030  001402                           BEQ     .+6             ;YES!!
040032  000137  041224                   JMP     1$              ;NO - BYPASS REST OF TEST
040036  113760  001226  000010           MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
040044  013737  001226  001240           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040052  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040056  016037  000012  001126           MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
040064  012737  000012  001122           MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040072  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
040076  012737  100000  001124           MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
040104  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
040112  042737  077777  001164           BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
040120  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
040126  001414                           BEQ     69$             ;BR IF OK
040130  013737  001126  001174           MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
040136  042737  100000  001174           BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
040144  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
040152  104010                           EMT     10
040154  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
040160  000240                  69$:     NOP
040162  005737  001250                   TST     CKERR           ;WAS ATTENTION SET FOR B??
040166  001402                           BEQ     .+6             ;YES!!
```

```
040170  000137  041224                  JMP     1$              ;NO - BYPASS REST OF TEST

                                ;SEIZE THE DRIVE THROUGH PORT A

040174  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
040202  013737  001224  001242          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
040210  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
040214  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
040222  013737  001226  001240          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040230  013737  001226  001244          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
040236  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
040244  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
040250  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
040256  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
040262  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
040270  001403                          BEQ     71$             ;BR IF IT IS
040272  104004                          EMT     4
040274  000137  041224                  JMP     1$              ;BYPASS REST OF THE SUBTEST
040300                          71$:
040300  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
040306  013737  001224  001240          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040314  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
040322  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
040330  012737  011700  001124          MOV     #MOL!PGM.DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
040336  013737  001124  001166          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
040344  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
040350  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
040356  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
040364  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
040372  001401                          BEQ     72$             ;BR IF THEY ARE
040374  104005                          EMT     5
040376  000240                  72$:    NOP

                                ;ISSUE NOP COMMAND TO PORT A

040400  012760  000001  000000          MOV     #1,RMCS1(R0)

                                ;VERIFY THAT ATTENTION FOR PORT A CLEARED

040406  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040412  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
040420  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040426  060037  001122                  ADD     R0,$BDADR        ADD RH/RM BASE ADDRESS
040432  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
040436  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
040444  042737  077777  001164          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
040452  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
040460  001414                          BEQ     73$             ;BR IF OK
040462  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
040470  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
040476  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
040504  104061                          EMT     61
040506  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
040512  000240                  73$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A
```

```
040514  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
040522  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040530  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

040536  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
040542  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
040550  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
040554  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
040562  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
040570  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
040576  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
040604  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
040612  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040620  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
040626  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
040634  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
040642  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
040650  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040656  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
040664  001006                          BNE     75$             ;BR IF NOT
040666  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
040672  001045                          BNE     77$             ;BR IF NOT
040674  104046                          EMT     46
040676  000137  041076                  JMP     79$             ;BYPASS THE REST OF THE CHECKS
040702  013737  001170  001126  75$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
040710  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040716  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
040724  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
040730  001414                          BEQ     76$             ;BR IF ZERO
040732  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040740  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
040746  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
040754  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
040760  001012                          BNE     77$             ;BR IF NOT
040762  012737   77777  001254  76$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
040770  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
040776  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
041004  104026                          EMT     26
041006  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
041014  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
041022  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
041030  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
041036  001401                          BEQ     78$             ;BR IF OK FROM PORT A.
041040  104007                          EMT     7
041042  013737  001172  001126  78$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
041050  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
041056  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
041064  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
041072  001401                          BEQ     79$             ;BR IF OK
041074  104007                          EMT     7
041076  000240                  79$:    NOP

                                ;VERIFY THAT ATTENTION FOR PORT B IS STIL SET

041100  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
```

```
041106  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041114  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
041120  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
041126  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041134  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
041140  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
041146  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
041154  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
041162  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
041170  001414                          BEQ     80$             ;BR IF OK
041172  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
041200  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
041206  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
041214  104062                          EMT     62
041216  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
041222  000240                  80$:    NOP
041224  000004                  1$:     SCOPE
```

```
;;*****************************************************************
;*TEST 31        RESET ATTENTION 'B' BY GO TEST
;*
;* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
;*       SEIZING PORT.
;*
;* A.    SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
;*       ATTENTION BITS ARE SET.
;*
;* B.    SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S
;*       INTO RMDS.
;*
;* C.    ISSUE A NOP COMMAND.
;*
;* D.    RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE
;*       ATTENTION BIT FOR PORT 'B' IS RESET, AND THE
;*       ATTENTION BIT FOR PORT 'A' IS STIL SET.
;*
;;*****************************************************************
041226                          TST31:
041226  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
041232  001406                          BEQ     2$              ;BR IF NOT
041234  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
041236  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
041242  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
041250  012737  041264  001106  2$:     MOV     #TEST31,$LPADR  ;SETUP SCOPE LOOP ADDRESS
041256  012737  041264  001110          MOV     #TEST31,$LPERR  ;SETUP ERROR LOOP ADDRESS
041264                          TEST31:
041264  112737  000031  001102          MOVB    #31,$TSTNM      ;MOVE #31 TO TEST NUMBER
041272  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
041276  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

;SET ATTENTION BITS FOR BOTH PORTS

041304  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
041312  005760  000012          66$:    TST     RMDS(R0)        ;MAKE SURE DRIVE  AVAILABLE
041316  001775                          BEQ     66$
```

```
041320  012760  177777  000014          MOV     #-1,RMER1(R0)    ;FORCE ERRORS
041326  005060  000014                  CLR     RMER1(R0)        ;CLEAR THE ERRORS
041332  013760  001226  000010          MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
041340  005760  000012          64$:    TST     RMDS(R0)         ;WAIT FOR DRIVE TO TIMEOUT
041344  001775                          BEQ     64$              ;BR IF DRIVE HASN'T TIMED OUT
041346  012760  177777  000014          MOV     #-1,RMER1(R0)    ;FORCE ERRORS ON PORT 65$
041354  005060  000014                  CLR     RMER1(R0)        ;CLEAR THE ERRORS
041360  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT "64$" AGAIN
041366  005760  000012          65$:    TST     RMDS(R0)         ;WAIT FOR DRIVE TO TIMEOUT
041372  001775                          BEQ     65$              ;BR IF DRIVE HASN'T TIMED OUT

                                ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

041374  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
041402  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041410  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
041414  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
041422  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041430  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
041434  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
041442  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
041450  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
041456  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
041464  001414                          BEQ     67$              ;BR IF OK
041466  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
041474  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
041502  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
041510  104010                          EMT     10
041512  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
041516  000240                  67$:    NOP
041520  005737  001250                  TST     CKERR            ;WAS ATTENTION SET FOR B??
041524  001402                          BEQ     .+6              ;YES!!
041526  000137  042720                  JMP     1$               ;NO - BYPASS REST OF TEST
041532  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
041540  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041546  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
041552  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
041560  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041566  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
041572  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
041600  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
041606  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
041614  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
041622  001414                          BEQ     69$              ;BR IF OK
041624  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
041632  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
041640  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
041646  104010                          EMT     10
041650  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
041654  000240                  69$:    NOP
041656  005737  001250                  TST     CKERR            ;WAS ATTENTION SET FOR A??
041662  001402                          BEQ     .+6              ;YES!!
041664  000137  042720                  JMP     1$               ;NO - BYPASS REST OF TEST

                                ;SEIZE THE DRIVE THROUGH PORT B

041670  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
```

```
041676  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
041704  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
041710  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
041716  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041724  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
041732  016037  000012  001126          MOV     RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
041740  010037  001122                  MOV     R0,$BDADR       ;RH/RM BASE ADDRESS
041744  062737  000012  001122          ADD     #RMDS,$BDADR    ;GENERATE BAD REGISTER ADDRESS
041752  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
041756  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
041764  001403                          BEQ     71$             ;BR IF IT IS
041766  104004                          EMT     4
041770  000137  042720                  JMP     1$              ;BYPASS REST OF THE SUBTEST
041774                          71$:
041774  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
042002  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042010  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
042016  042737  020001  001126          BIC     #OM!PIP,$BDDAT          ;CLEAR DONT CARE BITS
042024  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
042032  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
042040  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
042044  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
042052  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
042060  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
042066  001401                          BEQ     72$             ;BR IF THEY ARE
042070  104005                          EMT     5
042072  000240                  72$:    NOP

                                ;ISSUE NOP COMMAND TO PORT B

042074  012760  000001  000000          MOV     #1,RMCS1(R0)

                                ;VERIFY THAT ATTENTION FOR PORT B CLEARED

042102  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
042106  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
042114  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042122  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
042126  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
042132  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
042140  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
042146  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
042154  001414                          BEQ     73$             ;BR IF OK
042156  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
042164  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
042172  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
042200  104061                          EMT     61
042202  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
042206  000240                  73$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

042210  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
042216  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042224  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

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

```
042232  005037  001254                 CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
042236  012737  000012  0C1122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
042244  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
042250  012737  011700  001124          MOV     #MCL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
042256  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
042264  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
042272  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
042300  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
042306  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
042314  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
042322  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
042330  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
042336  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
042344  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
042352  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
042360  001006                          BNE     75$             ;BR IF NOT
042362  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
042366  001045                          BNE     77$             ;BR IF NOT
042370  104046                          EMT     46
042372  000137  042572                  JMP     79$             ;BYPASS THE REST OF THE CHECKS
042376  013737  001170  001126  75$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
042404  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
042412  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
042420  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM  ORT A.
042424  001414                          BEQ     76$             ;BR IF ZERO
042426  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
042434  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
042442  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
042450  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
042454  001012                          BNE     77$             ;BR IF NOT
042456  012737  177777  001254  76$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
042464  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
042472  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
042500  104026                          EMT     26
042502  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
042510  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
042516  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
042524  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
042532  001401                          BEQ     78$             ;BR IF OK FROM PORT A.
042534  104007                          EMT     7
042536  013737  001172  001126  78$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
042544  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
042552  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
042560  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
042566  001401                          BEQ     79$             ;BR IF OK
042570  104007                          EMT     7
042572  000240                  79$:    NOP

                                ;VERIFY THAT ATTENTION FOR PORT A IS STIL SET

042574  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
042602  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042610  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
042614  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
042622  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042630  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
```

```
042634  012737  100000  001124          MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
042642  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
042650  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
042656  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
042664  001414                          BEQ     80$             ;BR IF OK
042666  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
042674  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
042702  053737  001174  001124          BIS     $TMP4.$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
042710  104062                          EMT     62
042712  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
042716  000240                  80$:    NOP
042720  000004                  1$:     SCOPE
1054
1068
1069
                                ;;*****************************************************************
                                ;*TEST 32        TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT
                                ;*
                                ;*VERIFY THAT MASSBUS CLEAR  RESETS BOTH PORT'S ATTENTION BITS WHEN THE
                                ;*      DRIVE IS IN NEUTRAL.
                                ;*
                                ;*  A.  SET THE ATTENTION BITS FOR BOTH PORTS.
                                ;*
                                ;*  B.  VERIFY THAT THE DRIVE IS IN NEUTRAL.
                                ;*
                                ;*  C.  ISSUE A MASSBUS INIT.  VERIFY THAT BOTH ATTENTION BITS HAVE
                                ;*      RESET.
                                ;*
                                ;;*****************************************************************
042722                          TST32:
042722  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
042726  001406                          BEQ     2$              ;BR IF NOT
042730  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
042732  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
042736  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
042744  012737  042760  001106  2$:     MOV     #TEST32,$LPADR  ;SETUP SCOPE LOOP ADDRESS
042752  012737  042760  001110          MOV     #TEST32,$LPERR  ;SETUP ERROR LOOP ADDRESS
042760                          TEST32:
042760  112737  000032  001102          MOVB    #32,$TSTNM      ;MOVE #32 TO TEST NUMBER
042766  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
042772  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1070
1105
                                ;SET ATTENTION BITS FOR BOTH PORTS

043000  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
043006  005760  000012          66$:    TST     RMDS(R0)        ;MAKE SURE DRIVE  AVAILABLE
043012  001775                          BEQ     66$
043014  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS
043022  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
043026  013760  001226  000010          MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
043034  005760  000012          64$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
043040  001775                          BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
043042  012760  177777  000014          MOV     #-1,RMER1(R0)   ;FORCE ERRORS ON PORT 65$
043050  005060  000014                  CLR     RMER1(R0)       ;CLEAR THE ERRORS
043054  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT "64$" AGAIN
043062  005760  000012          65$:    TST     RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
043066  001775                          BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT
```

```
                                        ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

043070  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
043076  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043104  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
043110  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
043116  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043124  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
043130  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
043136  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
043144  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
043152  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
043160  001414                          BEQ     67$              ;BR IF OK
043162  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
043170  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
043176  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
043204  104010                          EMT     10
043206  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
043212  000240                  67$:    NOP
043214  005737  001250                  TST     CKERR            ;WAS ATTN BIT FOR PORT A SET ?
043220  001402                          BEQ     .+6              ;BR IF IT WAS
043222  000137  044176                  JMP     1$               ;BYPASS REST OF TEST IF NOT
043226  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
043234  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043242  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
043246  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
043254  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043262  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
043266  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
043274  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
043302  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
043310  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
043316  001414                          BEQ     69$              ;BR IF OK
043320  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
043326  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
043334  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
043342  104010                          EMT     10
043344  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
043350  000240                  69$:    NOP
043352  005737  001250                  TST     CKERR            ;WAS ATTN BIT FOR PORT B SET ?
043356  001402                          BEQ     .+6              ;BR IF IT WAS
043360  000137  044176                  JMP     1$               ;BYPASS REST OF TEST IF NOT

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

043364  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
043370  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
043376  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
043402  012737  111700  001124          MOV     #111700,$GDDAT   ;COMPARSION CONSTANT
043410  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
043416  016037  000012  001170          MOV     RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
043424  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
043432  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
043440  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043446  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
043454  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
043462  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
043470  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
043476  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043504  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
043512  001006                          BNE     71$             ;BR IF NOT
043514  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
043520  001045                          BNE     73$             ;BR IF NOT
043522  104046                          EMT     46
043524  000137  043710                  JMP     75$             ;BYPASS THE REST OF THE CHECKS
043530  013737  001170  001126  71$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
043536  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043544  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
043552  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
043556  001414                          BEQ     72$             ;BR IF ZERO
043560  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043566  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
043574  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
043602  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
043606  001012                          BNE     73$             ;BR IF NOT
043610  012737  177777  001254  72$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
043616  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
043624  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
043632  104026                          EMT     26
043634  013737  001170  001126  73$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
043642  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
043650  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
043656  001401                          BEQ     74$             ;BR IF OK FROM PORT A.
043660  104007                          EMT     7
043662  013737  001172  001126  74$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
043670  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
043676  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
043704  001401                          BEQ     75$             ;BR IF OK
043706  104007                          EMT     7
043710  000240                  75$:    NOP
043712  005737  001254                  TST     RELERR          ;WAS DRIVE IN NEUTRAL ?
043716  001402                          BEQ     .+6             ;BR IF IT WAS
043720  000137  044176                  JMP     1$              ;BYPASS RESET OF TEST

                                ;ISSUE THE MASSBUS INIT

043724  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;ISSUE A MASSBUS INIT

                                ;CHECK THE ATTENTION BITS OF BOTH PORTS

043732  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
043740  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043746  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
043752  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
043760  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043766  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
043772  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
043776  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
044004  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044012  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
044020  001414                          BEQ     76$             ;BR IF OK
044022  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
044030  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
```

```
044036  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
044044  104051                          EMT     51
044046  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
044052  000240                  76$:    NOP
044054  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
044062  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044070  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
044074  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
044102  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044110  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM 3ASE ADDRESS
044114  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
044120  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
044126  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044134  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
044142  001414                          BEQ     78$              ;BR IF OK
044144  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
044152  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
044160  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
044166  104051                          EMT     51
044170  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
044174  000240                  78$:    NOP
044176  000004                  1$:     SCOPE                    ;LOOP ?
1109
1123
1124
                                ;;*****************************************************************
                                ;*TEST 33        RESET ATTENTION 'A' & 'B' BY RMAS
                                ;*
                                ;*VERIFY THAT BOTH ATTENTION BITS CAN BE RESET BY WRITING THE
                                ;*APPROPRIATE BIT IN THE ATTENTION SUMMARY REGISTER.
                                ;*
                                ;* A.    SET THE ATTENTION BITS FOR BOTH PORTS.
                                ;*
                                ;* B.    VERIFY THE DRIVE IS IN NEUTRAL.
                                ;*
                                ;* C.    WRITE THE DRIVE'S ATTENTION BIT IN RMAS. VERIFY
                                ;*       THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMAS.
                                ;*
                                ;;*****************************************************************
044200                          TST33:
044200  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
044204  001406                          BEQ     2$               ;BR IF NOT
044206  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
044210  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
044214  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
044222  012737  044236  001106  2$:     MOV     #TEST33,$LPADR   ;SETUP SCOPE LOOP ADDRESS
044230  012737  044236  001110          MOV     #TEST33,$LPERR   ;SETUP ERROR LOOP ADDRESS
044236                          TEST33:
044236  112737  000033  001102          MOVB    #33,$TSTNM       ;MOVE #33 TO TEST NUMBER
044244  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
044250  012737  000002  001176          MOV     #2.,$TIMES       ;;DO 2. ITERATIONS
1125
1169
                                ;SET ATTENTION BITS FOR BOTH PORTS

044256  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT 64$
044264  005760  000012          66$:    TST     RMDS(R0)         ;MAKE SURE DRIVE  AVAILABLE
044270  001775                          BEQ     66$
```

```
044272  012760  177777  000014          MOV     #-1,RMER1(R0)    ;FORCE ERRORS
044300  005060  000014                  CLR     RMER1(R0)        ;CLEAR THE ERRORS
044304  013760  001226  000010          MOV     PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
044312  005760  000012          64$:    TST     RMDS(R0)         ;WAIT FOR DRIVE TO TIMEOUT
044316  001775                          BEQ     64$              ;BR IF DRIVE HASN'T TIMED OUT
044320  012760  177777  000014          MOV     #-1,RMER1(R0)    ;FORCE ERRORS ON PORT 65$
044326  005060  000014                  CLR     RMER1(R0)        ;CLEAR THE ERRORS
044332  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT "64$" AGAIN
044340  005760  000012          65$:    TST     RMDS(R0)         ;WAIT FOR DRIVE TO TIMEOUT
044344  001775                          BEQ     65$              ;BR IF DRIVE HASN'T TIMED OUT

                                ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
044346  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
044354  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044362  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
044366  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
044374  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044402  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
044406  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
044414  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
044422  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044430  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
044436  001414                          BEQ     67$              ;BR IF OK
044440  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
044446  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
044454  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
044462  104010                          EMT     10
044464  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
044470  000240                  67$:    NOP
044472  005737  001250                  TST     CKERR            ;WAS ATA SET FOR A??
044476  001402                          BEQ     .+6              ;YES - CONTINUE
044500  000137  045322                  JMP     1$               ;BYPASS REST OF TEST
044504  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
044512  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044520  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
044524  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
044532  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044540  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
044544  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
044552  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
044560  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044566  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
044574  001414                          BEQ     69$              ;BR IF OK
044576  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
044604  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
044612  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
044620  104010                          EMT     10
044622  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
044626  000240                  69$:    NOP
044630  005737  001250                  TST     CKERR            ;WAS ATA SET FOR B??
044634  001402                          BEQ     .+6              ;YES - CONTINUE
044636  000137  045322                  JMP     1$               ;BYPASS REST OF TEST

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

044642  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
044646  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
```

```
044654  060037  001122                  ADD     R0,$BDADR         ;ADD THE I/O BASE ADDRESS
044660  012737  111700  001124          MOV     #111700,$GDDAT    ;COMPARSION CONSTANT
044666  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
044674  016037  000012  001170          MOV     RMDS(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044702  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2         ;CLEAR DONT CARES
044710  013737  001170  001164          MOV     $TMP2,$TMP0       ;COPY IT INTO '$TMP0'
044716  042737  100100  001164          BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044724  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
044732  016037  000012  001172          MOV     RMDS(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
044740  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3         ;CLEAR DONT CARES
044746  013737  001172  001166          MOV     $TMP3,$TMP1       ;COPY IT INTO '$TMP1'
044754  042737  100100  001166          BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044762  023737  001164  001166          CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
044770  001006                          BNE     71$               ;BR IF NOT
044772  005737  001164                  TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
044776  001045                          BNE     73$               ;BR IF NOT
045000  104046                          EMT     46
045002  000137  045166                  JMP     75$               ;BYPASS THE REST OF THE CHECKS
045006  013737  001170  001126  71$:    MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
045014  013737  001226  001240          MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045022  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
045030  005737  001164                  TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
045034  001414                          BEQ     72$               ;BR IF ZERO
045036  013737  001224  001240          MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045044  013737  001172  001126          MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
045052  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
045060  005737  001166                  TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
045064  001012                          BNE     73$               ;BR IF NOT
045066  012737  177777  001254  72$:    MOV     #-1,RELERR        ;SET 'RELEASE ERROR' INDICATOR
045074  012760  000011  000000          MOV     #11,RMCS1(R0)     ;CLEAR THE DRIVE
045102  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
045110  104026                          EMT     26
045112  013737  001170  001126  73$:    MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
045120  013737  001224  001240          MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
045126  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;ALL BITS OK ?
045134  001401                          BEQ     74$               ;BR IF OK FROM PORT A.
045136  104007                          EMT     7
045140  013737  001172  001126  74$:    MOV     $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
045146  013737  001226  001240          MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
045154  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;SEE IF READ OK FROM PORT B.
045162  001401                          BEQ     75$               ;BR IF OK
045164  104007                          EMT     7
045166  000240                  75$:    NOP
045170  005737  001254                  TST     RELERR            ;WAS DRIVE IN NEUTRAL??
045174  001402                          BEQ     .+6               ;YES!!
045176  000137  045322                  JMP     1$                ;BYPASS REST OF TEST

                                ;WRITE THE ATTENTION BIT

045202  013760  001236  000016          MOV     ASR1,RMAS(R0)

                                ;VERIFY THAT BOTH ATTENTIONS ARE RESET BY READING RMAS
045210  016037  000016  001126          MOV     RMAS(R0),$BDDAT           ;GET ATTENTION SUMMARY
045216  033737  001236  001126          BIT     ASR1,$BDDAT               ;IS THE ATTENTION RESET ??
045224  001414                          BEQ     2$                        ;YES !!
045226  010037  001122                  MOV     R0,$BDADR                 ;SETUP REGISTER ADDRESS
045232  062737  000016  001122          ADD     #RMAS,$BDADR
```

```
045240  013737  001126  001124          MOV     $BDDAT,$GDDAT           ;SETUP EXPECTED DATA
045246  043737  001236  001124          BIC     ASR1,$GDDAT             ;RESET THIS DRIVES BIT
045254  104060                          EMT     60

045256                          2$:
                                ;WAIT FOR THE DRIVES TO RELEASE BY TIMEOUT

045256  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
045264  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045272  005760  000012          3$:     TST     RMDS(R0)        ;MAKE SURE DEVICE IS AVAILABLE
045276  001775                          BEQ     3$
045300  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
045306  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045314  005760  000012          4$:     TST     RMDS(R0)        ;MAKE SURE DEVICE IS AVAILABLE
045320  001775                          BEQ     4$
045322  000004                  1$:     SCOPE
1170
1183
1184                            ;;*****************************************************************
                                ;*TEST 34        PORT 'A' ALTERNATE ATTENTION PATH TEST
                                ;*
                                ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
                                ;*
                                ;*  A.  SET THE ATTENTION BIT FOR PORT 'A'.
                                ;*
                                ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
                                ;*      FOR THE DRIVE IS SET.
                                ;*
                                ;;*****************************************************************
045324                          TST34:
045324  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
045330  001406                          BEQ     2$              ;BR IF NOT
045332  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
045334  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
045340  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
045346  012737  045362  001106  2$:     MOV     #TEST34,$LPADR  ;SETUP SCOPE LOOP ADDRESS
045354  012737  045362  001110          MOV     #TEST34,$LPERR  ;SETUP ERROR LOOP ADDRESS
045362                          TEST34:
045362  112737  000034  001102          MOVB    #34,$TSTNM      ;MOVE #34 TO TEST NUMBER
045370  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
045374  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1185
1219
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

045402  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
045410  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
045414  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
045422  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
045430  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
045436  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
045442  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
045450  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
045456  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
045464  012760  177777  000014          MOV     #-1,RMER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT A
```

```
045472  005060  000014                  CLR     RMER1(R0)          ;CLEAR THE ERRORS
045476  113760  001226  000010          MOVB    PORTB,RMCS2(R0)     ;SELECT PORT B
045504  005760  000012          1$:     TST     RMDS(R0)           ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
045510  001775                          BEQ     1$                 ;BR IF STILL SEIZED BY PORT A
045512  012737  000016  001122          MOV     #RMAS,$BDADR       ;FORM ADDRESS OF ATTN REG IF ERROR
045520  060037  001122                  ADD     R0,$BDADR          ;ADD THE ADDRESS BASE
045524  013737  001236  001124          MOV     ASR1,$GDDAT        ;GOOD DATA FOR ERROR MESSAGE
045532  013737  001236  001166          MOV     ASR1,$TMP1         ;MAKE DATA COMPARE MASK
045540  005137  001166                  COM     $TMP1              ;COMPLEMENT IT
045544  012737  045600  001110          MOV     #2$,$LPERR         ;LOAD LOOP ON ERROR ADDRESS
045552  113760  001226  000010          MOVB    PORTB,RMCS2(R0)     ;SELECT PORT B
045560  013737  001226  001240          MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045566  013737  001226  001242          MOV     PORTB,SEIZPT       ;'SEIZED' PORT ADDRESS
045574  005060  000012                  CLR     RMDS(R0)           ;SEIZE THE DRIVE THROUGH PORT B
045600  016037  000016  001126  2$:     MOV     RMAS(R0),$BDDAT    ;GET THE CONTENTS OF THE ATTENTION REG
045606  013737  001126  001164          MOV     $BDDAT,$TMP0       ;PUT CONTENTS INTO WORKING LOCATION
045614  043737  001166  001164          BIC     $TMP1,$TMP0        ;CLEAR OTHER BITS
045622  023737  001124  001164          CMP     $GDDAT,$TMP0       ;SEE IF ATTN BIT FOR DRIVE SET
045630  001401                          BEQ     3$                 ;BR IF SET
045632  104053                          EMT     53
045634                          3$:
                                ;RELEASE THE DRIVE FROM PORT B

045634  113760  001226  000010          MOVB    PORTB,RMCS2(R0)     ;SELECT PORT B
045642  013737  001226  001240          MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045650  012760  000013  000000          MOV     #13,RMCS1(R0)      ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

045656  005037  001254                  CLR     RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
045662  012737  000012  001122          MOV     #RMDS,$BDADR       ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
045670  060037  001122                  ADD     R0,$BDADR          ;ADD THE I/O BASE ADDRESS
045674  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
045702  113760  001224  000010          MOVB    PORTA,RMCS2(R0)    ;SELECT PORT A.
045710  016037  000012  001170          MOV     RMDS(R0),$TMP2     ;GET THE DRIVE STATUS REGISTER FROM PORT A.
045716  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
045724  013737  001170  001164          MOV     $TMP2,$TMP0        ;COPY IT INTO '$TMP0'
045732  042737  100100  001164          BIC     #ATA!VV,$TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045740  113760  001226  000010          MOVB    PORTB,RMCS2(R0)    ;SELECT PORT B.
045746  016037  000012  001172          MOV     RMDS(R0),$TMP3     ;GET THE DRIVE STATUS REGISTER FROM PORT B.
045754  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
045762  013737  001172  001166          MOV     $TMP3,$TMP1        ;COPY IT INTO '$TMP1'
045770  042737  100100  001166          BIC     #ATA!VV,$TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045776  023737  001164  001166          CMP     $TMP0,$TMP1        ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046004  001006                          BNE     64$                ;BR IF NOT
046006  005737  001164                  TST     $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046012  001045                          BNE     66$                ;BR IF NOT
046014  104046                          EMT     46
046016  000137  046216                  JMP     68$                ;BYPASS THE REST OF THE CHECKS
046022  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT       ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046030  013737  001226  001240          MOV     PORTB,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046036  113760  001226  000010          MOVB    PORTB,RMCS2(R0)    ;SELECT PORT B.
046044  005737  001164                  TST     $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
046050  001414                          BEQ     65$                ;BR IF ZERO
046052  013737  001224  001240          MOV     PORTA,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046060  013737  001172  001126          MOV     $TMP3,$BDDAT       ;'BAD DATA' FOR ERROR TYPE OUT
046066  113760  001224  000010          MOVB    PORTA,RMCS2(R0)    ;SELECT PORT A.
```

```
046074  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
046100  001012                          BNE     66$             ;BR IF NOT
046102  012737  177777  001254  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
046110  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
046116  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
046124  104026                          EMT     26
046126  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
046134  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
046142  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
046150  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
046156  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
046160  104007                          EMT     7
046162  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
046170  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
046176  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
046204  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
046212  001401                          BEQ     68$             ;BR IF OK
046214  104007                          EMT     7
046216  000240                  68$:    NOP
046220  000004                          SCOPE                   ;LOOP ?
1220
1233
1234
                                ;;*****************************************************************
                                ;*TEST 35        PORT 'B' ALTERNATE ATTENTION PATH TEST
                                ;*
                                ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
                                ;*
                                ;*  A.  SET THE ATTENTION BIT FOR PORT 'B'.
                                ;*
                                ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
                                ;*      FOR THE DRIVE IS SET.
                                ;*
                                ;;*****************************************************************
046222                          TST35:
046222  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
046226  001406                          BEQ     2$              ;BR IF NOT
046230  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
046232  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
046236  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
046244  012737  046260  001106  2$:     MOV     #TEST35,$LPADR  ;SETUP SCOPE LOOP ADDRESS
046252  012737  046260  001110          MOV     #TEST35,$LPERR  ;SETUP ERROR LOOP ADDRESS
046260                          TEST35:
046260  112737  000035  001102          MOVB    #35,$TSTNM      ;MOVE #35 TO TEST NUMBER
046266  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
046272  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1235
1236
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

046300  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
046306  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
046312  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
046320  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
046326  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
046334  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
```

```
046340  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
046346  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
046354  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
046362  012760  177777  000014          MOV     #-1,RMER1(R0)    ;SET ERRORS TO FORCE ATTN BIT ON PORT B
046370  005060  000014                  CLR     RMER1(R0)        ;CLEAR THE ERRORS
046374  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
046402  005760  000012          1$:     TST     RMDS(R0)         ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
046406  001775                          BEQ     1$               ;BR IF STILL SEIZED BY PORT B
046410  012737  000016  001122          MOV     #RMAS,$BDADR     ;FORM ADDRESS OF ATTN REG IF ERROR
046416  060037  001122                  ADD     R0,$BDADR        ;ADD THE ADDRESS BASE
046422  013737  001236  001124          MOV     ASR1,$GDDAT      ;GOOD DATA FOR ERROR MESSAGE
046430  013737  001236  001166          MOV     ASR1,$TMP1       ;MAKE DATA COMPARE MASK
046436  005137  001166                  COM     $TMP1            ;COMPLEMENT IT
046442  012737  046476  001110          MOV     #2$,$LPERR       ;LOAD LOOP ON ERROR ADDRESS
046450  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
046456  013737  001224  001240          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046464  013737  001224  001242          MOV     PORTA,SEIZPT     ;'SEIZED' PORT ADDRESS
046472  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT A
046476  016037  000016  001126  2$:     MOV     RMAS(R0),$BDDAT  ;GET THE CONTENTS OF THE ATTENTION REG
046504  013737  001126  001164          MOV     $BDDAT,$TMP0     ;PUT CONTENTS INTO WORKING LOCATION
046512  043737  001166  001164          BIC     $TMP1,$TMP0      ;CLEAR OTHER BITS
046520  023737  001124  001164          CMP     $GDDAT,$TMP0     ;SEE IF ATTN BIT FOR DRIVE SET
046526  001401                          BEQ     3$               ;BR IF SET
046530  104053                          EMT     53
046532                          3$:
                                ;RELEASE THE DRIVE FROM PORT A

046532  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
046540  013737  001224  001240          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046546  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

046554  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
046560  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
046566  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
046572  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
046600  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
046606  016037  000012  001170          MOV     RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
046614  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
046622  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
046630  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046636  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
046644  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046652  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
046660  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
046666  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046674  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046702  001006                          BNE     64$              ;BR IF NOT
046704  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046710  001045                          BNE     66$              ;BR IF NOT
046712  104046                          EMT     46
046714  000137  047114                  JMP     68$              ;BYPASS THE REST OF THE CHECKS
046720  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046726  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046734  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
046742  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
```

```
046746  001414                          BEQ     65$                 ;BR IF ZERO
046750  013737  001224  001240          MOV     PORTA,PTNBR         ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046756  013737  001172  001126          MOV     $TMP3,$BDDAT        ;'BAD DATA' FOR ERROR TYPE OUT
046764  113760  001224  000010          MOVB    PORTA,RMCS2(R0)     ;SELECT PORT A.
046772  005737  001166                  TST     $TMP1               ;SEE IF STATUS EQ ZERO FROM PORT B.
046776  001012                          BNE     66$                 ;BR IF NOT
047000  012737  177777  001254  65$:    MOV     #-1,RELERR          ;SET 'RELEASE ERROR' INDICATOR
047006  012760  000011  000000          MOV     #11,RMCS1(R0)       ;CLEAR THE DRIVE
047014  012760  000013  000000          MOV     #13,RMCS1(R0)       ;RELEASE THE DRIVE
047022  104026                          EMT     26
047024  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT        ;LOOK FOR BIT FAILURES WHEN RMDS READ
047032  013737  001224  001240          MOV     PORTA,PTNBR         ;CHANGE PORT NUMBER
047040  042737  100000  001126          BIC     #ATA,$BDDAT         ;DON'T CHECK THE ATTN BIT
047046  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;ALL BITS OK ?
047054  001401                          BEQ     67$                 ;BR IF OK FROM PORT A.
047056  104007                          EMT     7
047060  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT        ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
047066  013737  001226  001240          MOV     PORTB,PTNBR         ;CHANGE PORT NUMBER
047074  042737  100000  001126          BIC     #ATA,$BDDAT         ;DON'T CHECK THE ATTN BIT
047102  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;SEE IF READ OK FROM PORT B.
047110  001401                          BEQ     68$                 ;BR IF OK
047112  104007                          EMT     7
047114  000240                  68$:    NOP
047116  000004                          SCOPE                       ;LOOP ?
1237
1254
1255                            ;;*****************************************************************
                                ;*TEST 36        SET ATTENTION 'A' BY COMMMAND TEST
                                ;*
                                ;*
                                ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
                                ;*      COMMAND.
                                ;*
                                ;*  A.  ISSUE A OFFSET COMMAND THROUGH PORT 'A'.
                                ;*
                                ;*  B.  WAIT FOR THE OFFSET COMMAND TO COMPLETE ('DRY' TO BECOME
                                ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
                                ;*      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
                                ;*
                                ;*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
                                ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
047120                          TST36:
047120  005737  001300                  TST     KYBCTL              ;PERFORMING ONLY SINGLE TEST ?
047124  001406                          BEQ     2$                  ;BR IF NOT
047126  100002                          BPL     1$                  ;BR IF JUST ENTERED TEST
047130  000137  003062                  JMP     EXEC                ;RETURN & GET NEXT TEST NUMBER
047134  012737  177777  001300  1$:     MOV     #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
047142  012737  047156  001106  2$:     MOV     #TEST36,$LPADR      ;SETUP SCOPE LOOP ADDRESS
047150  012737  047156  001110          MOV     #TEST36,$LPERR      ;SETUP ERROR LOOP ADDRESS
047156                          TEST36:
047156  112737  000036  001102          MOVB    #36,$TSTNM          ;MOVE #36 TO TEST NUMBER
047164  012706  001100                  MOV     #STACK,SP           ;LOAD THE STACK POINTER
047170  012737  000012  001176          MOV     #10.,$TIMES         ;;DO 10. ITERATIONS
1256
1285
```

```
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

047176  113760  001224  000010           MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
047204  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
047210  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
047216  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
047224  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
047232  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
047236  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
047244  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
047252  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
047260  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047266  013737  001224  001242          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS

                                        ;DO A OFFSET THROUGH PORT A

047274  012760  000015  000000          MOV     #15,RMCS1(R0)   ;ISSUE A OFFSET INSTRUCTION THROUGH PORT A

                                        ;WAIT FOR DRIVE TO FINISH

047302  032760  000200  000012          BIT     #DRY,RMDS(R0)   ;WAIT FOR DRIVE TO FINISH
047310  001774                          BEQ     .-6             ;BR IF NOT FINISHED

                                        ;CONFIRM THAT ATTENTION IS SET FOR PORT A

047312  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
047316  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
047324  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
047332  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
047336  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
047344  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
047352  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
047360  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
047366  001414                          BEQ     64$             ;BR IF OK
047370  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
047376  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
047404  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
047412  104032                          EMT     32
047414  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
047420  000240                  64$:    NOP

                                        ;RELEASE THE DRIVE FROM PORT A

047422  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
047430  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047436  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

047444  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
047450  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
047456  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
047462  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
047470  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
047476  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
047504  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
047512  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
047520  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047526  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
047534  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
047542  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
047550  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
047556  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047564  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
047572  001006                          BNE     66$              ;BR IF NOT
047574  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
047600  001045                          BNE     68$              ;BR IF NOT
047602  104046                          EMT     46
047604  000137  047770                  JMP     70$              ;BYPASS THE REST OF THE CHECKS
047610  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047616  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047624  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
047632  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
047636  001414                          BEQ     67$              ;BR IF ZERO
047640  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047646  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
047654  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
047662  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
047666  001012                          BNE     68$              ;BR IF NOT
047670  012737  177777  001254  67$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
047676  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
047704  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
047712  104026                          EMT     26
047714  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
047722  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
047730  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
047736  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
047740  104007                          EMT     7
047742  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
047750  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
047756  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
047764  001401                          BEQ     70$              ;BR IF OK
047766  104007                          EMT     7
047770  000240                  70$:    NOP
047772  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
050000  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

050006  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
050012  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
050020  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
050026  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
050032  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
050036  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
050044  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
050052  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
050060  001414                          BEQ     71$              ;BR IF OK
050062  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
050070  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
050076  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
050104  104032                          EMT     32
050106  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
050112  000240                  71$:    NOP
```

```
      050114  000004                          SCOPE                   ;LOOP ?
 1286
 1302
 1303                                 ;;*****************************************************************
                                      ;*TEST 37       SET ATTENTION 'B' BY COMMMAND TEST
                                      ;*
                                      ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
                                      ;*      COMMAND.
                                      ;*
                                      ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
                                      ;*
                                      ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
                                      ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND
                                      ;*      THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
                                      ;*
                                      ;*  C.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
                                      ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                      ;*
                                      ;;*****************************************************************
      050116                          TST37:
      050116  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
      050122  001406                          BEQ     2$              ;BR IF NOT
      050124  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
      050126  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
      050132  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
      050140  012737  050154  001106  2$:     MOV     #TEST37,$LPADR  ;SETUP SCOPE LOOP ADDRESS
      050146  012737  050154  001110          MOV     #TEST37,$LPERR  ;SETUP ERROR LOOP ADDRESS
      050154                          TEST37:
      050154  112737  000037  001102          MOVB    #37,$TSTNM      ;MOVE #37 TO TEST NUMBER
      050162  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
      050166  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 1304
 1305
                                      ;CLEAR ATTENTION BITS FOR BOTH PORTS

      050174  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
      050202  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
      050206  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
      050214  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
      050222  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
      050230  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
      050234  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
      050242  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
      050250  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
      050256  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      050264  013737  001226  001242          MOV     PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS

                                      ;DO A OFFSET THROUGH PORT B

      050272  012760  000015  000000          MOV     #15,RMCS1(R0)   ;ISSUE A OFFSET INSTRUCTION THROUGH PORT B

                                      ;WAIT FOR DRIVE TO FINISH

      050300  032760  000200  000012          BIT     #DRY,RMDS(R0)   ;WAIT FOR DRIVE TO FINISH
      050306  001774                          BEQ     .-6             ;BR IF NOT FINISHED
```

```
                                        ;CONFIRM THAT ATTENTION IS SET FOR PORT B

050310  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
050314  016037  000012  001126                  MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
050322  012737  000012  001122                  MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
050330  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
050334  012737  100000  001124                  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
050342  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
050350  042737  077777  001164                  BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
050356  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
050364  001414                                  BEQ     64$             ;BR IF OK
050366  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
050374  042737  100000  001174                  BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
050402  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
050410  104032                                  EMT     32
050412  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
050416  000240                          64$:    NOP

                                        ;RELEASE THE DRIVE FROM PORT B

050420  113760  001226  000010                  MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
050426  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050434  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

050442  005037  001254                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
050446  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
050454  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
050460  012737  011700  001124                  MOV     #MOL PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
050466  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
050474  016037  000012  001170                  MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
050502  042737  024001  001170                  BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
050510  013737  001170  001164                  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
050516  042737  100100  001164                  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050524  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
050532  016037  000012  001172                  MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
050540  042737  024001  001172                  BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
050546  013737  001172  001166                  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
050554  042737  100100  001166                  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050562  023737  001164  001166                  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
050570  001006                                  BNE     66$             ;BR IF NOT
050572  005737  001164                          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
050576  001045                                  BNE     68$             ;BR IF NOT
050600  104046                                  EMT     46
050602  000137  050766                          JMP     70$             ;BYPASS THE REST OF THE CHECKS
050606  013737  001170  001126          66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
050614  013737  001226  001240                  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050622  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
050630  005737  001164                          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
050634  001414                                  BEQ     67$             ;BR IF ZERO
050636  013737  001224  001240                  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050644  013737  001172  001126                  MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
050652  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
050660  005737  001166                          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
050664  001012                                  BNE     68$             ;BR IF NOT
050666  012737  177777  001254          67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
```

```
050674  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
050702  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
050710  104026                          EMT     26
050712  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
050720  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
050726  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
050734  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
050736  104007                          EMT     7
050740  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
050746  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
050754  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
050762  001401                          BEQ     70$              ;BR IF OK
050764  104007                          EMT     7
050766  000240                  70$:    NOP
050770  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
050776  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

051004  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
051010  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
051016  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051024  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
051030  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
051034  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
051042  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
051050  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
051056  001414                          BEQ     71$              ;BR IF OK
051060  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
051066  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
051074  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
051102  104032                          EMT     32
051104  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
051110  000240                  71$:    NOP

051112  000004                          SCOPE                    ;LOOP ?
1309
1318
1319                            ;;*****************************************************************
1320                            ;*
                                ;*VERIFY THAT A CHANGE IN UNIT READY SETS THE ATTENTION
                                ;*      FOR BOTH PORTS.
                                ;*
                                ;*THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.
                                ;*
1321                            ;;*****************************************************************
1322
1331
1332                            ;;*****************************************************************
1333                            ;*
                                ;*VERIFY THAT ATTENTION SETS WHEN THE DRIVE SWITCHES AFTER
                                ;*BEING RELEASED.
                                ;*
                                ;*THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"
                                ;*
1334                            ;;*****************************************************************
1335
```

```
1356
1357
                                        ;;*****************************************************************
                                        ;*TEST 40        PORT 'A' SET VOLUME VALID TEST
                                        ;*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
                                        ;*
                                        ;* A.   WITH PORT 'A' SELECTED, RESET AND SET 'MUR' IN RMMR1,
                                        ;*      USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
                                        ;*      IS SEIZED AND THAT 'VOLUME VALID' IS RESET AND
                                        ;*      ATTENTION IS SET.
                                        ;*
                                        ;* B.   ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
                                        ;*      COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
                                        ;*      VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
                                        ;*      IS SET.
                                        ;*
                                        ;* C.   RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR
                                        ;*      PORT 'B'. VERIFY THAT ATTENTION IS STIL SET AND THAT
                                        ;*      VOLUME VALID IS STIL RESET.
                                        ;*
                                        ;* D.   ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
                                        ;*      PORT 'B' THEN RELEASE PORT 'B'.
                                        ;;*****************************************************************
051114                                  TST40.
051114  005737  001300                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
051120  001406                                  BEQ     2$              ;BR IF NOT
051122  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
051124  000137  003062                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
051130  012737  177777  001300          1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
051136  012737  051152  001106          2$:     MOV     #TEST40,$LPADR  ;SETUP SCOPE LOOP ADDRESS
051144  012737  051152  001110                  MOV     #TEST4C,$LPERR  ;SETUP ERROR LOOP ADDRESS
051152                                  TEST40:
051152  112737  000040  001102                  MOVB    #40,$TSTNM      ;MOVE #40 TO TEST NUMBER
051160  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
051164  012737  000012  001176                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1358
1399
                                        ;SEIZE PORT A BY WRITING THE MAINTENANCE REGISTER, RMMR1. SET
                                        ;AND RESET 'MUR' TO CAUSE VOLUME VALID TO RESET AND ATTENTION TO SET.

                                        ;SEIZE THE DRIVE THROUGH PORT A

051172  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
051200  013737  001224  001242                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
051206  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;WRITE DMD INTO RMMR1
051214  013737  001226  001244                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
051222  012760  001001  000024                  MOV     #DMD!MUR,RMMR1(R0)      ;SET UNIT READY
051230  012760  000000  000024                  MOV     #0,RMMR1(R0)            ;RESET DIAGNOSTIC MODE

                                        ;VERIFY THAT ATA = 1,VV = 0 FOR PORT A
051236  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
051242  016037  000012  001126                  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
051250  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051256  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
051262  012737  100000  001124                  MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
051270  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
051276  042737  077777  001164                  BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
051304  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
```

```
051312  001414                          BEQ     66$             ;BR IF OK
051314  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
051322  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
051330  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
051336  104064                          EMT     64
051340  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
051344  000240                  66$:    NOP
051346  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
051352  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
051360  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051366  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
051372  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
051376  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
051404  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
051412  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
051420  001414                          BEQ     68$             ;BR IF OK
051422  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
051430  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
051436  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
051444  104065                          EMT     65
051446  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
051452  000240                  68$:    NOP

                                ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A
051454  012760  000011  000000          MOV     #11,RMCS1(R0)           ;DRIVE CLEAR
051462  012760  000021  000000          MOV     #21,RMCS1(R0)           ;READ IN PRESET

                                ;VERIFY ATA = 0 AND VV = 1 FOR PORT A
051470  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
051474  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
051502  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051510  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
051514  012737  000100  001124          MOV     #VV,$GDDAT  ;WHAT REGISTER SHOULD BE
051522  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
051530  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
051536  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
051544  001414                          BEQ     70$             ;BR IF OK
051546  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
051554  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
051562  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
051570  104013                          EMT     13
051572  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
051576  000240                  70$:    NOP
051600  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
051604  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
051612  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051620  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
051624  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
051632  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
051640  042737  024007  001164          BIC     #^C153770,$TMP0  ;SAVE SPECIFIED BITS
051646  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
051654  001414                          BEQ     72$             ;BR IF OK
051656  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
051664  042737  153770  001174          BIC     #153770,$TMP4   ;CLEAR THE MASKED BITS
051672  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
051700  104010                          EMT     10
051702  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
051706  000240                          72$:    NOP

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

                                        ;RELEASE THE DRIVE FROM PORT A

051710  113760  001224  000010                  MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
051716  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051724  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

051732  005037  001254                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
051736  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
051744  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
051750  012737  011600  001124                  MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
051756  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
051764  016037  000012  001170                  MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
051772  042737  024001  001170                  BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
052000  013737  001170  001164                  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
052006  042737  100100  001164                  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052014  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
052022  016037  000012  001172                  MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
052030  042737  024001  001172                  BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
052036  013737  001172  001166                  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
052044  042737  100100  001166                  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052052  023737  001164  001166                  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052060  001006                                  BNE     74$             ;BR IF NOT
052062  005737  001164                          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052066  001037                                  BNE     76$             ;BR IF NOT
052070  104046                                  EMT     46
052072  000137  052256                          JMP     78$             ;BYPASS THE REST OF THE CHECKS
052076  013737  001170  001126  74$:            MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052104  013737  001226  001240                  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052112  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
052120  005737  001164                          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
052124  001414                                  BEQ     75$             ;BR IF ZERO
052126  013737  001224  001240                  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052134  013737  001172  001126                  MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
052142  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
052150  005737  001166                          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
052154  001004                                  BNE     76$             ;BR IF NOT
052156  012737  177777  001254  75$:            MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
052164  104022                                  EMT     22
052166  013737  001170  001126  76$:            MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
052174  013737  001224  001240                  MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
052202  042737  100100  001126                  BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
052210  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
052216  001401                                  BEQ     77$             ;BR IF OK FROM PORT A.
052220  104007                                  EMT     7
052222  013737  001172  001126  77$:            MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
052230  013737  001226  001240                  MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
052236  042737  100100  001126                  BIC     #ATA!VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
052244  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
052252  001401                                  BEQ     78$             ;BR IF OK
052254  104007                                  EMT     7
```

```
052256  000240                  78$:    NOP
052260  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
052266  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052274  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
052300  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
052306  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052314  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
052320  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
052326  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
052334  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
052342  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
052350  001414                          BEQ     79$              ;BR IF OK
052352  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
052360  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
052366  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
052374  104064                          EMT     64
052376  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
052402  000240                  79$:    NOP
052404  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
052410  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
052416  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052424  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
052430  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
052434  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
052442  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
052450  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
052456  001414                          BEQ     81$              ;BR IF OK
052460  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
052466  042737  000100  001174          BIC     #VV,$TMP4        ;CLEAR THE MASKED BITS
052474  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
052502  104065                          EMT     65
052504  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
052510  000240                  81$:    NOP

                                ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B,
                                ;THEN RELEASE PORT B
052512  012760  000011  000000          MOV     #11,RMCS1(R0)    ;DRIVE CLEAR
052520  012760  000021  000000          MOV     #21,RMCS1(R0)    ;READ IN PRESET
                                ;RELEASE THE DRIVE FROM PORT B

052526  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
052534  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052542  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

052550  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
052554  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
052562  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
052566  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
052574  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
052602  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
052610  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
052616  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
052624  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052632  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
052640  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
052646  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
052654  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
052662  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052670  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052676  001006                          BNE     83$              ;BR IF NOT
052700  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052704  001037                          BNE     85$              ;BR IF NOT
052706  104046                          EMT     46
052710  000137  053074                  JMP     87$              ;BYPASS THE REST OF THE CHECKS
052714  013737  001170  001126  83$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052722  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052730  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B.
052736  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
052742  001414                          BEQ     84$              ;BR IF ZERO
052744  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052752  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
052760  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A.
052766  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
052772  001004                          BNE     85$              ;BR IF NOT
052774  012737  177777  001254  84$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
053002  104022                          EMT     22
053004  013737  001170  001126  85$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
053012  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
053020  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
053026  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
053034  001401                          BEQ     86$              ;BR IF OK FROM PORT A.
053036  104007                          EMT     7
053040  013737  001172  001126  86$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
053046  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
053054  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
053062  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
053070  001401                          BEQ     87$              ;BR IF OK
053072  104007                          EMT     7
053074  000240                  87$:    NOP

053076  000004                  50$:    SCOPE
1400
1421
1422
                                ;*****************************************************************
                                ;*TEST 41        PORT 'B' SET VOL  E VALID TEST
                                ;*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
                                ;*
                                ;* A.    WITH PORT 'B' SELECTED, RESET AND SET "MUR" IN RMMR1,
                                ;*       USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
                                ;*       IS SEIZED AND THAT "VOLUME VALID" IS RESET AND
                                ;*       ATTENTION IS SET.
                                ;*
                                ;* B.    ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
                                ;*       COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
                                ;*       VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
                                ;*       IS SET.
                                ;*
                                ;* C.    RELEASE THE DRIVE FROM PORT 'B' AND SELECT THE DRIVE FOR
                                ;*       PORT 'A'. VERIFY THAT ATTENTION IS STIL SET AND THAT
                                ;*       VOLUME VALID IS STIL RESET.
                                ;*
                                ;* D.    ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
```

```
                                        ;*      PORT 'A' THEN RELEASE PORT 'A'.
                                        ;;*****************************************************************
053100                                  TST41:
053100  005737  001300                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
053104  001406                                  BEQ     2$              ;BR IF NOT
053106  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
053110  000137  003062                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
053114  012737  177777  001300  1$:             MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
053122  012737  053136  001106  2$:             MOV     #TEST41,$LPADR  ;SETUP SCOPE LOOP ADDRESS
053130  012737  053136  001110                  MOV     #TEST41,$LPERR  ;SETUP ERROR LOOP ADDRESS
053136                                  TEST41:
053136  112737  000041  001102                  MOVB    #41,$TSTNM      ;MOVE #41 TO TEST NUMBER
053144  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
053150  012737  000012  001176                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 1423
 1424
                                        ;SEIZE PORT B BY WRITING THE MAINTENANCE REGISTER, RMMR1. SET
                                        ;AND RESET 'MUR' TO CAUSE VOLUME VALID TO RESET AND ATTENTION TO SET.

                                        ;SEIZE THE DRIVE THROUGH PORT B

053156  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
053164  013737  001226  001242                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
053172  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;WRITE DMD INTO RMMR1
053200  013737  001224  001244                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
053206  012760  001001  000024                  MOV     #DMD!MUR,RMMR1(R0)      ;SET UNIT READY
053214  012760  000000  000024                  MOV     #0,RMMR1(R0)            ;RESET DIAGNOSTIC MODE

                                        ;VERIFY THAT ATA = 1,VV = 0 FOR PORT B
053222  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
053226  016037  000012  001126                  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
053234  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053242  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
053246  012737  100000  001124                  MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
053254  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
053262  042737  077777  001164                  BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
053270  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
053276  001414                                  BEQ     66$             ;BR IF OK
053300  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
053306  042737  100000  001174                  BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
053314  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
053322  104064                                  EMT     64
053324  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
053330  000240                          66$:    NOP
053332  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
053336  016037  000012  001126                  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
053344  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053352  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
053356  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
053362  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
053370  042737  177677  001164                  BIC     #^CVV,$TMP0     ;SAVE SPECIFIED BITS
053376  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
053404  001414                                  BEQ     68$             ;BR IF OK
053406  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
053414  042737  000100  001174                  BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
053422  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
053430  104065                                  EMT     65
```

```
053432  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
053436  000240                  68$:    NOP

                                ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B
053440  012760  000011  000000          MOV     #11,RMCS1(R0)           ;DRIVE CLEAR
053446  012760  000021  000000          MOV     #21,RMCS1(R0)           ;READ IN PRESET

                                ;VERIFY ATA = 0 AND VV = 1 FOR PORT B
053454  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
053460  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
053466  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053474  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
053500  012737  000100  001124          MOV     #VV,$GDDAT  ;WHAT REGISTER SHOULD BE
053506  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
053514  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
053522  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
053530  001414                          BEQ     70$             ;BR IF OK
053532  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
053540  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
053546  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
053554  104013                          EMT     13
053556  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
053562  000240                  70$:    NOP
053564  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
053570  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
053576  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053604  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
053610  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
053616  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
053624  042737  024007  001164          BIC     #^C153770,$TMP0  ;SAVE SPECIFIED BITS
053632  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
053640  001414                          BEQ     72$             ;BR IF OK
053642  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
053650  042737  153770  001174          BIC     #153770,$TMP4   ;CLEAR THE MASKED BITS
053656  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
053664  104010                          EMT     10
053666  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
053672  000240                  72$:    NOP

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

                                ;RELEASE THE DRIVE FROM PORT B

053674  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
053702  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
053710  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

053716  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
053722  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
053730  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
053734  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
053742  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
053750  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
053756  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
```

```
053764  013737  001170  001164          MOV     $TMP2,$TMP0       ;COPY IT INTO '$TMP0'
053772  042737  100100  001164          BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054000  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
054006  016037  000012  001172          MOV     RMDS(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054014  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3         ;CLEAR DONT CARES
054022  013737  001172  001166          MOV     $TMP3,$TMP1       ;COPY IT INTO '$TMP1'
054030  042737  100100  001166          BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054036  023737  001164  001166          CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054044  001006                          BNE     74$               ;BR IF NOT
054046  005737  001164                  TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054052  001037                          BNE     76$               ;BR IF NOT
054054  104046                          EMT     46
054056  000137  054242                  JMP     78$               ;BYPASS THE REST OF THE CHECKS
054062  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054070  013737  001226  001240          MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054076  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
054104  005737  001164                  TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
054110  001414                          BEQ     75$               ;BR IF ZERO
054112  013737  001224  001240          MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054120  013737  001172  001126          MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
054126  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
054134  005737  001166                  TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
054140  001004                          BNE     76$               ;BR IF NOT
054142  012737  177777  001254  75$:    MOV     #-1,RELERR        ;SET 'RELEASE ERROR' INDICATOR
054150  104022                          EMT     22
054152  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
054160  013737  001224  001240          MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
054166  042737  100100  001126          BIC     #ATA!VV,$BDDAT    ;DON'T CHECK ATTN BIT OR VV BIT
054174  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;ALL BITS OK ?
054202  001401                          BEQ     77$               ;BR IF OK FROM PORT A.
054204  104007                          EMT     7
054206  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
054214  013737  001226  001240          MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
054222  042737  100100  001126          BIC     #ATA!VV,$BDDAT    ;DON'T CHECK ATTN BIT OR VV BIT
054230  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;SEE IF READ OK FROM PORT B.
054236  001401                          BEQ     78$               ;BR IF OK
054240  104007                          EMT     7
054242  000240                  78$:    NOP
054244  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
054252  013737  001224  001240          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054260  005037  001250                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
054264  016037  000012  001126          MOV     RMDS(R0),$BDDAT   ;GET CONTENTS OF RMDS
054272  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054300  060037  001122                  ADD     R0,$BDADR         ;ADD RH/RM BASE ADDRESS
054304  012737  100000  001124          MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
054312  013737  001126  001164          MOV     $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
054320  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
054326  023737  001124  001164          CMP     $GDDAT,$TMP0      ;COMPARE THE BITS
054334  001414                          BEQ     79$               ;BR IF OK
054336  013737  001126  001174          MOV     $BDDAT,$TMP4      ;COPY 'BAD DATA'
054344  042737  100000  001174          BIC     #ATA,$TMP4        ;CLEAR THE MASKED BITS
054352  053737  001174  001124          BIS     $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
054360  104064                          EMT     64
054362  005137  001250                  COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
054366  000240                  79$:    NOP
054370  005037  001250                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
054374  016037  000012  001126          MOV     RMDS(R0),$BDDAT   ;GET CONTENTS OF RMDS
```

```
054402  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054410  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
054414  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
054420  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
054426  042737  177677  001164          BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
054434  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
054442  001414                          BEQ     81$             ;BR IF OK
054444  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
054452  042737  000100  001174          BIC     #VV,$TMP4       ;CLEAR THE MASKED BITS
054460  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
054466  104065                          EMT     65
054470  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
054474  000240                  81$:    NOP

                                ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A,
                                ;THEN RELEASE PORT A
054476  012760  000011  000000          MOV     #11,RMCS1(R0)   ;DRIVE CLEAR
054504  012760  000021  000000          MOV     #21,RMCS1(R0)   ;READ IN PRESET
                                ;RELEASE THE DRIVE FROM PORT A

054512  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
054520  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054526  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

054534  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
054540  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
054546  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
054552  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
054560  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
054566  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054574  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
054602  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
054610  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054616  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
054624  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054632  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
054640  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
054646  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054654  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054662  001006                          BNE     83$             ;BR IF NOT
054664  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054670  001037                          BNE     85$             ;BR IF NOT
054672  104046                          EMT     46
054674  000137  055060                  JMP     87$             ;BYPASS THE REST OF THE CHECKS
054700  013737  001170  001126  83$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054706  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054714  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
054722  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
054726  001414                          BEQ     84$             ;BR IF ZERO
054730  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054736  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
054744  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
054752  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
054756  001004                          BNE     85$             ;BR IF NOT
054760  012737  177777  001254  84$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
```

```
        054766  104022                          EMT     22
        054770  013737  001170  001126  85$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
        054776  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
        055004  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
        055012  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
        055020  001401                          BEQ     86$              ;BR IF OK FROM PORT A.
        055022  104007                          EMT     7
        055024  013737  001172  001126  86$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
        055032  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
        055040  042737  100100  001126          BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
        055046  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
        055054  001401                          BEQ     87$              ;BR IF OK
        055056  104007                          EMT     7
        055060  000240                  87$:    NOP

        055062  000004                  50$:    SCOPE
   1429
   1443
   1444                                 ;;*****************************************************************
                                        ;*TEST 42        TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
                                        ;*
                                        ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
                                        ;*
                                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                        ;*
                                        ;* B.  WRITE 1'S INTO RMER1 THROUGH PORT 'A' TO FORCE AN ATTENTION.
                                        ;*
                                        ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
                                        ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND NOT SET FOR
                                        ;*      PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
                                        ;*
                                        ;;*****************************************************************
        055064                          TST42:
        055064  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
        055070  001406                          BEQ     2$               ;BR IF NOT
        055072  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
        055074  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
        055100  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
        055106  012737  055122  001106  2$:     MOV     #TEST42,$LPADR   ;SETUP SCOPE LOOP ADDRESS
        055114  012737  055122  001110          MOV     #TEST42,$LPERR   ;SETUP ERROR LOOP ADDRESS
        055122                          TEST42:
        055122  112737  000042  001102          MOVB    #42,$TSTNM       ;MOVE #42 TO TEST NUMBER
        055130  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
        055134  012737  000002  001176          MOV     #2.,$TIMES       ;;DO 2. ITERATIONS
   1445
   1496
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

        055142  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT #A
        055150  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE
        055154  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
        055162  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
        055170  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT #B
        055176  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
        055202  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
        055210  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
```

```
                                        ;SEIZE THE DRIVE THROUGH PORT A

055216  113760  001224  000010           MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
055224  013737  001224  001242           MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
055232  005060  000012                   CLR     RMDS(R0)        ;WRITE RMDS
055236  013737  001226  001244           MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;FORCE AN ATTENTION BY SETTING ERRORS.

055244  012760  177777  000014           MOV     #-1,RMER1(R0)   ;SET ERROR BITS

                                        ;START THE TIMER

055252  005037  001256                   CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
055256  012737  003720  001260           MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS
055264  113760  001226  000010           MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
055272  013737  001226  001210           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;WAIT FOR DRIVE TO TIMEOUT

055300  005760  000012          1$:      TST     RMDS(R0)        ;WAIT FOR THE DRIVE TO BE RELEASED
055304  001004                           BNE     2$              ;BR IF DRIVE RELEASED
055306  005737  001260                   TST     WATCH           ;WATCH AT ZERO ?
055312  001372                           BNE     1$              ;BR IF NOT
055314  104036                           EMT     36
055316                          2$:
055316  113760  001224  000010           MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
055324  013737  001224  001240           MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

055332  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
055336  016037  000012  001126           MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
055344  012737  000012  001122           MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055352  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
055356  012737  040000  001124           MOV     #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
055364  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
055372  042737  137777  001164           BIC     #^CERR,$TMP0  ;SAVE SPECIFIED BITS
055400  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
055406  001414                           BEQ     66$             ;BR IF OK
055410  013737  001126  001174           MOV     $BDDAT,$TMP4     COPY 'BAD DATA'
055416  042737  040000  001174           BIC     #ERR,$TMP4      ;CLEAR THE MASKED BITS
055424  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
055432  104023                           EMT     23
055434  005137  001250                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
055440  000240                  66$:     NOP

                                        ;THE ERROR REGISTER SHOULD CONTAIN 1'S

055442  005037  001250                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
055446  016037  000014  001126           MOV     RMER1(R0),$BDDAT  ;GET CONTENTS OF RMER1
055454  012737  000014  001122           MOV     #RMER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055462  060037  001122                   ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
055466  012737  177777  001124           MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
055474  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
055502  001403                           BEQ     68$             ;BR IF OK
055504  104010                           EMT     10
```

```
055506  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
055512  000240                  68$:    NOP

                                ;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

055514  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
055520  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
055526  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055534  060037  001122                  ADD     R0,$BDADR       ;AD  RH/RM BASE ADDRESS
055540  012737  100000  001124          MOV     #ATA,$BDDAT     ;WHAT REGISTER SHOULD BE
055546  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
055554  042737  077777  001164          BIC     #^C<ATA>,$TMP0  ;SAVE SPECIFIED BITS
055562  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
055570  001414                          BEQ     70$             ;BR IF OK
055572  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
055600  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
055606  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
055614  104041                          EMT     41
055616  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
055622  000240                  70$:    NOP

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

055624  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
055630  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
055636  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
055642  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
055650  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
055656  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
055664  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
055672  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
055700  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055706  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
055714  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
055722  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
055730  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
055736  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055744  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
055752  001006                          BNE     72$             ;BR IF NOT
055754  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
055760  001045                          BNE     74$             ;BR IF NOT
055762  104046                          EMT     46
055764  000137  056164                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
055770  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
055776  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056004  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
056012  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
056016  001414                          BEQ     73$             ;BR IF ZERO
056020  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056026  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
056034  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
056042  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
056046  001012                          BNE     74$             ;BR IF NOT
056050  012737  177777  001254  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
056056  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
056064  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
056072  104026                          EMT     26
```

```
056074  013737  001170  001126  74$:    MOV     $TMP2,$FDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
056102  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
056110  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
056116  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
056124  001401                          BEQ     75$              ;BR IF OK FROM PORT A.
056126  104007                          EMT     7
056130  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
056136  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
056144  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
056152  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
056160  001401                          BEQ     76$              ;BR IF OK
056162  104007                          EMT     7
056164  000240          76$:            NOP

                        ;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

056166  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
056174  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
056202  005037  001250                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
056206  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
056214  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056222  060037  001122                  ADD     R0,$BDADR        ;ADD RH/RM BASE ADDRESS
056226  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
056232  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
056240  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
056246  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
056254  001414                          BEQ     77$              ;BR IF OK
056256  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
056264  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
056272  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
056300  104052                          EMT     52
056302  005137  001250                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
056306  000240          77$:            NOP

                        ;CLEAR ATTENTION BIT FOR PORT A

056310  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT #A
056316  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE
056322  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
056330  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
056336  000004          3$:             LOOPE                    ;LOOP ?
1497
1511
1512                    ;;*****************************************************************
                        ;*TEST 43        TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
                        ;*
                        ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
                        ;*
                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                        ;*
                        ;*  B   WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
                        ;*
                        ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
                        ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
                        ;*      PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
                        ;*
                        ;;*****************************************************************
```

```
056340                                  TST43:
056340  005737  001300                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
056344  001406                                  BEQ     2$              ;BR IF NOT
056346  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
056350  000137  003062                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
056354  012737  177777  001300  1$:             MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
056362  012737  056376  001106  2$:             MOV     #TEST43,$LPADR  ;SETUP SCOPE LOOP ADDRESS
056370  012737  056376  001110                  MOV     #TEST43,$LPERR  ;SETUP ERROR LOOP ADDRESS
056376                                  TEST43:
056376  112737  000043  001102                  MOVB    #43,$TSTNM      ;MOVE #43 TO TEST NUMBER
056404  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
056410  012737  000002  001176                  MOV     #2.,$TIMES      ;;DO 2. ITERATIONS
 1513
 1514
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

056416  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
056424  005060  000012                          CLR     RMDS(R0)        ;SEIZE THE DRIVE
056430  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
056436  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
056444  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
056452  005060  000012                          CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
056456  012760  000011  000000                  MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
056464  012760  000013  000000                  MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT B

056472  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
056500  013737  001226  001242                  MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
056506  005060  000012                          CLR     RMDS(R0)        ;WRITE RMDS
056512  013737  001224  001244                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;FORCE AN ATTENTION BY SETTING ERRORS.

056520  012760  177777  000014                  MOV     #-1,RMER1(R0)   ;SET ERROR BITS

                                        ;START THE TIMER

056526  005037  001256                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
056532  012737  003720  001260                  MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS
056540  113760  001224  000010                  MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
056546  013737  001224  001240                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;WAIT FOR DRIVE TO TIMEOUT

056554  005760  000012                  1$:     TST     RMDS(R0)        ;WAIT FOR THE DRIVE TO BE RELEASED
056560  001004                                  BNE     2$              ;BR IF DRIVE RELEASED
056562  005737  001260                          TST     WATCH           ;WATCH AT ZERO ?
056566  001372                                  BNE     1$              ;BR IF NOT
056570  104036                                  EMT     36
056572                                  2$:
056572  113760  001226  000010                  MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
056600  013737  001226  001240                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

056606  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
056612  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
056620  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056626  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
056632  012737  040000  001124          MOV     #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
056640  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
056646  042737  137777  001164          BIC     #^CERR,$TMP0  ;SAVE SPECIFIED BITS
056654  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
056662  001414                          BEQ     66$             ;BR IF OK
056664  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
056672  042737  040000  001174          BIC     #ERR,$TMP4      ;CLEAR THE MASKED BITS
056700  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
056706  104023                          EMT     23
056710  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
056714  000240                  66$:    NOP

                                ;THE ERROR REGISTER SHOULD CONTAIN 1'S

056716  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
056722  016037  000014  001126          MOV     RMER1(R0),$BDDAT  ;GET CONTENTS OF RMER1
056730  012737  000014  001122          MOV     #RMER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056736  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
056742  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
056750  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
056756  001403                          BEQ     68$             ;BR IF OK
056760  104010                          EMT     10
056762  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
056766  000240                  68$:    NOP

                                ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

056770  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
056774  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
057002  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057010  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
057014  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
057022  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
057030  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
057036  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
057044  001414                          BEQ     70$             ;BR IF OK
057046  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
057054  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
057062  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
057070  104041                          EMT     41
057072  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
057076  000240                  70$:    NOP

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

057100  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE E  OR ' INDICATOR
057104  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
057112  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
057116  012737  051700  001124          MOV     #51700,$GDDAT  ;COMPARSION CONSTANT
057124  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
057132  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
057140  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
057146  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
057154  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```

```
057162  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
057170  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
057176  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
057204  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
057212  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
057220  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
057226  001006                          BNE     72$             ;BR IF NOT
057230  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
057234  001045                          BNE     74$             ;BR IF NOT
057236  104046                          EMT     46
057240  000137  057440                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
057244  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
057252  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057260  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
057266  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
057272  001414                          BEQ     73$             ;BR IF ZERO
057274  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057302  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
057310  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
057316  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
057322  001012                          BNE     74$             ;BR IF NOT
057324  012737  177777  001254  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
057332  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
057340  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
057346  104026                          EMT     26
057350  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
057356  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
057364  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
057372  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
057400  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
057402  104007                          EMT     7
057404  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
057412  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
057420  042737  100000  001126          BIC     #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
057426  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
057434  001401                          BEQ     76$             ;BR IF OK
057436  104007                          EMT     7
057440  000240                  76$:    NOP

                                ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

057442  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
057450  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
057456  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
057462  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
057470  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057476  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
057502  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
057506  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
057514  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
057522  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
057530  001414                          BEQ     77$             ;BR IF OK
057532  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
057540  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
057546  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
057554  104052                          EMT     52
057556  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
057562  000240                    77$:    NOP

                                  ;CLEAR ATTENTION BIT FOR PORT B

057564  113760  001226  000010            MOVB    PORTB,RMCS2(R0)  ;SELECT PORT #B
057572  005060  000012                    CLR     RMDS(R0)         ;SEIZE THE DRIVE
057576  012760  000011  000000            MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
057604  012760  000013  000000            MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
057612  000004                    3$:     SCOPE                    ;LOOP ?
1515
1531
1532
                                  ;;*****************************************************************
                                  ;*TEST 44        PORT 'A' RETRIGGER BY DEMAND TEST
                                  ;*
                                  ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
                                  ;*
                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                  ;*
                                  ;*  B.  WAIT 500 MS AND READ RMDS THROUGH PORT 'A'.
                                  ;*
                                  ;*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
                                  ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
                                  ;*
                                  ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
                                  ;*      BIT IS SET.
                                  ;*
                                  ;;****************** ***********************************************
057614                            TST44:
057614  005737  001300                    TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
057620  001406                            BEQ     2$               ;BR IF NOT
057622  100002                            BPL     1$               ;BR IF JUST ENTERED TEST
057624  000137  003062                    JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
057630  012737  177777  001300    1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
057636  012737  057652  001106    2$:     MOV     #TEST44,$LPADR   ;SETUP SCOPE LOOP ADDRESS
057644  012737  057652  001110            MOV     #TEST44,$LPERR   ;SETUP ERROR LOOP ADDRESS
057652                            TEST44:
057652  112737  000044  001102            MOVB    #44,$TSTNM       ;MOVE #44 TO TEST NUMBER
057660  012706  001100                    MOV     #STACK,SP        ;LOAD THE STACK POINTER
057664  012737  000002  001176            MOV     #2.,$TIMES       ;;DO 2. ITERATIONS
1533
1568
                                  ;CLEAR ATTENTION BITS FOR BOTH PORTS

057672  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
057700  005060  000012                    CLR     RMDS(R0)         ;SEIZE THE DRIVE
057704  012760  000011  000000            MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
057712  012760  000013  000000            MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
057720  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
057726  005060  000012                    CLR     RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
057732  012760  000011  000000            MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
057740  012760  000013  000000            MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE

                                  ;SEIZE THE DRIVE THROUGH PORT A

057746  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
057754  013737  001224  001242            MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
057762  005060  000012                    CLR     RMDS(R0)         ;WRITE RMDS
```

```
057766  013737  001226  001244          MOV     PORTB,OPPRT     :'OPPOSITE' PORT ADDRESS

                                ;START THE TIMER

057774  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
060000  012`37  000764  001260          MOV     #500.,WATCH     ;SET WATCH TO TIM. MS
060006  005737  001260          1$:     TST     WATCH           ;WATCH EQUAL TO ZERO
060012  001375                          BNE     1$              ;BR IF NOT

                                ;START THE TIMER

060014  005037  001256                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
060020  012737  003720  001260          MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS

                                ;RETRIGGER THE TIMEOUT ONE-SHOT

060026  005760  000012                  TST     RMDS(R0)        ;RETRIGGER THE ONE-SHOT
060032  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
060040  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060046  005760  000012          2$:     TST     RMDS(R0)        ;WAIT FOR TIMEOUT
060052  001004                          BNE     3$              ;BR IF TIMEOUT OCCURRED
060054  005737  001260                  TST     WATCH           ;WATCH EQUAL TO ZERO ?
060060  001372                          BNE     2$              ;BR IF NOT
060062  104036                          EMT     36
060064  013737  001256  001276  3$:     MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060072  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
060076  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
060104  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
060110  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
060116  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
060124  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
060132  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
060140  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
060146  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060154  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
060162  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
060170  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
060176  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
060204  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060212  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
060220  001006                          BNE     66$             ;BR IF NOT
060222  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
060226  001045                          BNE     68$             ;BR IF NOT
060230  104046                          EMT     46
060232  000137  060416                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
060236  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
060244  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060252  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
060260  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
060264  001414                          BEQ     67$             ;BR IF ZERO
060266  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060274  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
060302  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
060310  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
```

```
060314  001012                          BNE     68$             ;BR IF NOT
060316  012737  177777  001254  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
060324  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
060332  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
060340  104022                          EMT     22
060342  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
060350  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
060356  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
060364  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
060366  104007                          EMT     7
060370  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
060376  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
060404  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
060412  001401                          BEQ     70$             ;BR IF OK
060414  104007                          EMT     7
060416  000240                  70$:    NOP

                                ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

060420  023737  001276  001264          CMP     TIMES,TIMEAP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
060426  003004                          BGT     4$              ;BR IF GREATER
060430  023737  001276  001266          CMP     TIMES,TIMEAM    ;MEASURED TIME LESS THAN -25% TOLERANCE
060436  002001                          BGE     .+4             ;BR IF NOT
060440                          4$:
060440  104025                          EMT     25
060442  000004                          SCOPE                   ;LOOP ?
1569
1585
1586                            ;;*****************************************************************
                                ;*TEST 45        PORT 'B' RETRIGGER BY DEMAND TEST
                                ;*
                                ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WAIT 500 MS AND WRITE 0'B INTO RMDS THROUGH PORT 'A'.
                                ;*
                                ;*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
                                ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
                                ;*
                                ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
                                ;*      BIT IS SET.
                                ;*
                                ;;*****************************************************************
060444                          TST45:
060444  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
060450  001406                          BEQ     2$              ;BR IF NOT
060452  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
060454  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
060460  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
060466  012737  060502  001106  2$:     MOV     #TEST45,$LPADR  ;SETUP SCOPE LOOP ADDRESS
060474  012737  060502  001110          MOV     #TEST45,$LPERR  ;SETUP ERROR LOOP ADDRESS
060502                          TEST45:
060502  112737  000045  001102          MOVB    #45,$TSTNM      ;MOVE #45 TO TEST NUMBER
060510  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
060514  012737  000002  001176          MOV     #2.,$TIMES      ;;DO 2. ITERATIONS
1587
```

```
1588
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

060522  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
060530  005060  000012                    CLR     RMDS(R0)        ;SEIZE THE DRIVE
060534  012760  000011  000000            MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
060542  012760  000013  000000            MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
060550  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
060556  005060  000012                    CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
060562  012760  000011  000000            MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
060570  012760  000013  000000            MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT B

060576  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
060604  013737  001226  001242            MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
060612  005060  000012                    CLR     RMDS(R0)        ;WRITE RMDS
060616  013737  001224  001244            MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                        ;START THE TIMER

060624  005037  001256                    CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
060630  012737  000764  001260            MOV     #500.,WATCH     ;SET WATCH TO TIM. MS
060636  005737  001260          1$:       TST     WATCH           ;WATCH EQUAL TO ZERO
060642  001375                            BNE     1$              ;BR IF NOT

                                        ;START THE TIMER

060644  005037  001256                    CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
060650  012737  003720  001260            MOV     #2000.,WATCH    ;SET WATCH TO 2000. MS

                                        ;RETRIGGER THE TIMEOUT ONE-SHOT

060656  005760  000012                    TST     RMDS(R0)        ;RETRIGGER THE ONE-SHOT
060662  113760  001224  000010            MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
060670  013737  001224  001240            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060676  005760  000012          2$:       TST     RMDS(R0)        ;WAIT FOR TIMEOUT
060702  001004                            BNE     3$              ;BR IF TIMEOUT OCCURRED
060704  005737  001260                    TST     WATCH           ;WATCH EQUAL TO ZERO ?
060710  001372                            BNE     2$              ;BR IF NOT
060712  104036                            EMT     36
060714  013737  001256  001276  3$:       MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060722  005037  001254                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
060726  012737  000012  001122            MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
060734  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
060740  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
060746  113760  001224  000010            MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
060754  016037  000012  001170            MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
060762  042737  024001  001170            BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
060770  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
060776  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061004  113760  001226  000010            MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
061012  016037  000012  001172            MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061020  042737  024001  001172            BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
```

```
061026  013737  001172  001166          MOV     $TMP3,$TMP1       ;COPY IT INTO '$TMP1'
061034  042737  100100  001166          BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061042  023737  001164  001166          CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
061050  001006                          BNE     66$               ;BR IF NOT
061052  005737  001164                  TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
061056  001045                          BNE     68$               ;BR IF NOT
061060  104046                          EMT     46
061062  000137  061246                  JMP     70$               ;BYPASS THE REST OF THE CHECKS
061066  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
061074  013737  001226  001240          MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061102  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B.
061110  005737  001164                  TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
061114  001414                          BEQ     67$               ;BR IF ZERO
061116  013737  001224  001240          MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061124  013737  001172  001126          MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
061132  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A.
061140  005737  001166                  TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
061144  001012                          BNE     68$               ;BR IF NOT
061146  012737  177777  001254  67$:    MOV     #-1,RELERR        ;SET 'RELEASE ERROR' INDICATOR
061154  012760  000011  000000          MOV     #11,RMCS1(R0)     ;CLEAR THE DRIVE
061162  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE
061170  104022                          EMT     22
061172  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
061200  013737  001224  001240          MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
061206  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;ALL BITS OK ?
061214  001401                          BEQ     69$               ;BR IF OK FROM PORT A.
061216  104007                          EMT     7
061220  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
061226  013737  001226  001240          MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
061234  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;SEE IF READ OK FROM PORT B.
061242  001401                          BEQ     70$               ;BR IF OK
061244  104007                          EMT     7
061246  000240                  70$:    NOP

                                ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

061250  023737  001276  001272          CMP     TIMES,TIMEBP      ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
061256  003004                          BGT     4$                ;BR IF GREATER
061260  023737  001276  001274          CMP     TIMES,TIMEBM      ;MEASURED TIME LESS THAN -25% TOLERANCE
061266  002001                          BGE     .+4               ;BR IF NOT
061270                          4$:
061270  104025                          EMT     25
061272  000004                          SCOPE                     ;LOOP ?
1589
1610
1611                            ;;*****************************************************************
                                ;*TEST 46        PORT 'A'  TIMEOUT/RELEASE TEST
                                ;*
                                ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
                                ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'A'.
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B'.  VERIFY THAT THE DRIVE
                                ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
```

```
                                        ;*      SET FOR PORT 'B'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
                                        ;*
                                        ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS
                                        ;*      BEEN RELEASED.
                                        ;*
                                        ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;*****************************************************************
     061274                             TST46:
     061274  005737  001300                     TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
     061300  001406                             BEQ     2$              ;BR IF NOT
     061302  100002                             BPL     1$              ;BR IF JUST ENTERED TEST
     061304  000137  003062                     JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
     061310  012737  177777  001300     1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
     061316  012737  061332  001106     2$:     MOV     #TEST46,$LPADR  ;SETUP SCOPE LOOP ADDRESS
     061324  012737  061332  001110             MOV     #TEST46,$LPERR  ;SETUP ERROR LOOP ADDRESS
     061332                             TEST46:
     061332  112737  000046  001102             MOVB    #46,$TSTNM      ;MOVE #46 TO TEST NUMBER
     061340  012706  001100                     MOV     #STACK,SP       ;LOAD THE STACK POINTER
     061344  012737  000002  001176             MOV     #2.,$TIMES      ;;DO 2. ITERATIONS
  1612
  1656
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

     061352  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
     061360  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE
     061364  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     061372  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
     061400  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
     061406  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
     061412  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     061420  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT B

     061426  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
     061434  013737  001226  001242             MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
     061442  005060  000012                     CLR     RMDS(R0)        ;WRITE RMDS
     061446  013737  001224  001244             MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
     061454  113760  001224  000010             MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
     061462  013737  001224  001240             MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;SET REQUEST THROUGH PORT A

     061470  005060  000012                     CLR     RMDS(R0)        ;SET REQUEST FOR PORT A
     061474  113760  001226  000010             MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
     061502  013737  001226  001240             MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;RELEASE THE DRIVE THROUGH PORT B

     061510  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE DRIVE THROUGH PORT B

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

     061516  013737  001264  001260             MOV     TIMEAP,WATCH    ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
```

```
                                        ;VERIFY THAT THE DRIVE IS SEIZED BY PORT A

061524  005037  001250                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
061530  016037  000012  001126                  MOV     RMDS(R0),$BDDAT  ;GET CONTENTS OF RMDS
061536  012737  000012  001122                  MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
061544  060037  001122                          ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
061550  005037  001124                          CLR     $GDDAT          ;WHAT REG'STER SHOULD BE
061554  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
061562  001403                                  BEQ     66$             ;BR IF OK
061564  104031                                  EMT     31
061566  005137  001250                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
061572  000240                          66$:    NOP
061574  005737  001250                          TST     CKERR           ;REGISTER OK ?
061600  001402                                  BEQ     .+6             ;BR IF OK
061602  000137  062156                          JMP     1$              ;BYPASS REST OF TEST IF NOT


                                        ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
061606  005737  001260                          TST     WATCH           ;WATCH EQUAL ZERO ?
061612  001375                                  BNE     .-4             ;BR IF NOT

                                        ;CONFIRM THAT THE DRIVE HAS TIMED OUT

                                        ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

061614  005037  001254                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
061620  012737  000012  001122                  MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
061626  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
061632  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
061640  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
061646  016037  000012  001170                  MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
061654  042737  024001  001170                  BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
061662  013737  001170  001164                  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
061670  042737  100100  001164                  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061676  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
061704  016037  000012  001172                  MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061712  042737  024001  001172                  BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
061720  013737  001172  001166                  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
061726  042737  100100  001166                  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061734  023737  001164  001166                  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
061742  001006                                  BNE     68$             ;BR IF NOT
061744  005737  001164                          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
061750  001045                                  BNE     70$             ;BR IF NOT
061752  104046                                  EMT     46
061754  000137  062154                          JMP     72$             ;BYPASS THE REST OF THE CHECKS
061760  013737  001170  001126          68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
061766  013737  001226  001240                  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061774  113760  001226  000010                  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
062002  005737  001164                          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
062006  001414                                  BEQ     69$             ;BR IF ZERO
062010  013737  001224  001240                  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062016  013737  001172  001126                  MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
062024  113760  001224  000010                  MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
062032  005737  001166                          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
062036  001012                                  BNE     70$             ;BR IF NOT
062040  012737  177777  001254          69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
062046  012737  000011  000000                  MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
```

```
062054  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
062062  104035                          EMT     35
062064  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
062072  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
062100  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
062106  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
062114  001401                          BEQ     71$              ;BR IF OK FROM PORT A.
062116  104007                          EMT     7
062120  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
062126  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
062134  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
062142  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
062150  001401                          BEQ     72$              ;BR IF OK
062152  104007                          EMT     7
062154  000240                  72$:    NOP
062156  000004                  1$:     SCOPE                    ;LOOP ?
1657
1675
1676
                                ;;*****************************************************************
                                ;*TEST 47        PORT 'B'  TIMEOUT/RELEASE TEST
                                ;*
                                ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
                                ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'B'.
                                ;*
                                ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A'.  VERIFY THAT THE DRIVE
                                ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE '/TA' BIT DID NOT
                                ;*      SET FOR PORT 'A'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
                                ;*
                                ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS
                                ;*      BEEN RELEASED.
                                ;*
                                ;;*****************************************************************
062160                          TST47:
062160  005737  001300                  TST     KYBCTL           ;PERFORMING ONLY SINGLE TEST ?
062164  001406                          BEQ     2$               ;BR IF NOT
062166  100002                          BPL     1$               ;BR IF JUST ENTERED TEST
062170  000137  003062                  JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
062174  012737  177777  001300  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
062202  012737  062216  001106  2$:     MOV     #TEST47,$LPADR   ;SETUP SCOPE LOOP ADDRESS
062210  012737  062216  001110          MOV     #TEST47,$LPERR   ;SETUP ERROR LOOP ADDRESS
062216                          TEST47:
062216  112737  000047  001102          MOVB    #47,$TSTNM       ;MOVE #47 TO TEST NUMBER
062224  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
062230  012737  000002  001176          MOV     #2.,$TIMES       ;;DO 2. ITERATIONS
1677
1678
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

062236  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT #A
062244  005060  000012                  CLR     RMDS(R0)         ;SEIZE THE DRIVE
062250  012760  000011  000000          MOV     #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
062256  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
062264  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT #B
```

```
062272  005060  000012                  CLR     RMDS(R0)          ;SEIZE THE DRIVE THROUGH PORT 'B'
062276  012760  000011  000000          MOV     #11,RMCS1(R0)     ;ISSUE DRIVE CLEAR
062304  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT A

062312  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
062320  013737  001224  001242          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
062326  005060  000012                  CLR     RMDS(R0)          ;WRITE RMDS
062332  013737  001226  001244          MOV     PORTB,OPPRT       ;'OPPOSITE' PORT ADDRESS
062340  113760  001226  000010          MOVB    PORTB,RMCS2(R0)   ;SELECT PORT B
062346  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;SET REQUEST THROUGH PORT B

062354  005060  000012                  CLR      MDS(R0)          ;SET REQUEST FOR PORT B
062360  113760  001224  000010          MOVB    PORTA,RMCS2(R0)   ;SELECT PORT A
062366  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;RELEASE THE DRIVE THROUGH PORT A

062374  012760  000013  000000          MOV     #13,RMCS1(R0)     ;RELEASE DRIVE THROUGH PORT A

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

062402  013737  001272  001260          MOV     TIMEBP,WATCH      ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%

                                ;VERIFY THAT THE DRIVE IS SEIZED BY PORT B

062410  005037  001250                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
062414  016037  000012  001126          MOV     RMDS(R0),$BDDAT   ;GET CONTENTS OF RMDS
062422  012737  000012  001122          MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
062430  060037  001122                  ADD     R0,$BDADR         ;ADD RH/RM BASE ADDRESS
062434  005037  001124                  CLR     $GDDAT            ;WHAT REGISTER SHOULD BE
062440  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;IS THE REGISTER OK ?
062446  001403                          BEQ     66$               ;BR IF OK
062450  104031                          EMT     31
062452  005137  001250                  COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
062456  000240                  66$:    NOP
062460  005737  001250                  TST     CKERR             ;REGISTER OK ?
062464  001402                          BEQ     .+6               ;BR IF OK
062466  000137  063042                  JMP     1$                ;BYPASS REST OF TEST IF NOT


                                ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
062472  005737  001260                  TST     WATCH             ;WATCH EQUAL ZERO ?
062476  001375                          BNE     .-4               ;BR IF NOT

                                ;CONFIRM THAT THE DRIVE HAS TIMED OUT

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

062500  005037  001254                  CLR     RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
062504  012737  000012  001122          MOV     #RMDS,$BDADR      ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
062512  060037  001122                  ADD     R0,$BDADR         ;ADD THE I/O BASE ADDRESS
062516  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
062524  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
```

```
062532  016037  000012  001170          MOV     RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
062540  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2        ;CLEAR DONT CARES
062546  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
062554  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062562  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
062570  016037  000012  001172          MOV     RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
062576  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3        ;CLEAR DONT CARES
062604  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
062612  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062620  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
062626  [illegible]006                  BNE     68$              ;BR IF NOT
062630  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
062634  001045                          BNE     70$              ;BR IF NOT
062636  104046                          EMT     46
062640  000137  063040                  JMP     72$              ;BYPASS THE REST OF THE CHECKS
062644  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
062652  013737  001226  001240          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062660  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
062666  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
062672  001414                          BEQ     69$              ;BR IF ZERO
062674  013737  001224  001240          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062702  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
062710  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
062716  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
062722  001012                          BNE     70$              ;BR IF NOT
062724  012737  177777  001254  69$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
062732  012760  000011  000000          MOV     #11,RMCS1(R0)    ;CLEAR THE DRIVE
062740  012760  000013  000000          MOV     #13,RMCS1(R0)    ;RELEASE THE DRIVE
062746  104035                          EMT     35
062750  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
062756  013737  001224  001240          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
062764  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
062772  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
063000  001401                          BEQ     71$              ;BR IF OK FROM PORT A.
063002  104007                          EMT     7
063004  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
063012  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
063020  042737  100000  001126          BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
063026  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
063034  001401                          BEQ     72$              ;BR IF OK
063036  104007                          EMT     7
063040  000240                  72$:    NOP
063042  000004                  1$:     SCOPE                    ;LOOP ?
1679
1704
1705                            ;;*****************************************************************
                                ;*TEST 50        PORT 'A' SEIZE ACCESS TEST
                                ;*
                                ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.
                                ;*
                                ;*  C.  READ RMER1, RMER2 THROUGH PORT 'B'.  VERIFY THAT PORT
                                ;*      'B' SEES 0'S FROM EACH OF THESE REGISTERS.
                                ;*
```

```
                                        ;*  D.  CLEAR RMER1, RMER2 THROUGH PORT 'A'.
                                        ;*
                                        ;*  E.  WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.  VERIFY THAT
                                        ;*      PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
                                        ;*
                                        ;*  F.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE HAS
                                        ;*      SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
                                        ;*      SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
                                        ;*
                                        ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
     063044                             TST50:
     063044  005737  001300                     TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
     063050  001406                             BEQ     2$              ;BR IF NOT
     063052  100002                             BPL     1$              ;BR IF JUST ENTERED TEST
     063054  000137  003062                     JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
     063060  012737  177777  001300  1$:        MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
     063066  012737  063102  001106  2$:        MOV     #TEST50,$LPADR  ;SETUP SCOPE LOOP ADDRESS
     063074  012737  063102  001110             MOV     #TEST50,$LPERR  ;SETUP ERROR LOOP ADDRESS
     063102                             TEST50:
     063102  112737  000050  001102             MOVB    #50,$TSTNM      ;MOVE #50 TO TEST NUMBER
     063110  012706  001100                     MOV     #STACK,SP       ;LOAD THE STACK POINTER
     063114  012737  000012  001176             MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
  1706
  1745
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

     063122  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
     063130  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE
     063134  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     063142  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
     063150  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
     063156  005060  000012                     CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
     063162  012760  000011  000000             MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
     063170  012760  000013  000000             MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                        ;SEIZE THE DRIVE THROUGH PORT A

     063176  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
     063204  013737  001224  001242             MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
     063212  005060  000012                     CLR     RMDS(R0)        ;WRITE RMDS
     063216  013737  001226  001244             MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
     063224  012760  177777  000014             MOV     #-1,RMER1(R0)   ;LOAD 1'S INTO RMER1 THROUGH PORT A
     063232  012760  177777  000042             MOV     #-1,RMER2(R0)   ;LOAD 1'S INTO RMER2 THROUGH PORT A
     063240  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
     063246  013737  001226  001240             MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     063254  004737  064114                     JSR     PC,TST50B       ;CHECK THE REGISTERS THROUGH PORT B
     063260  113760  001224  000010             MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
     063266  013737  001224  001240             MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     063274  005060  000042                     CLR     RMER2(R0)       ;CLEAR RMER2 ON PORT A
     063300  005060  000014                     CLR     RMER1(R0)       ;CLEAR RMER1 ON PORT A
     063304  013760  001236  000016             MOV     ASR1,RMAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT A
     063312  113760  001226  000010             MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
     063320  013737  001226  001240             MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
     063326  012760  177777  000014             MOV     #-1,RMER1(R0)   ;LOAD 1'S INTO RMER1 THROUGH PORT B
```

```
063334  012760  177777  000042          MOV     #-1,RMER2(R0)    ;LOAD 1'S INTO RMER2 THROUGH PORT B
063342  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
063350  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063356  004737  064114                  JSR     PC,TST50B        ;CHECK THE REGISTERS THROUGH PORT A

                                ;RELEASE THE DRIVE FROM PORT A

063362  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
063370  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063376  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

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

063404  005037  001254                  CLR     RELERR           ;CLEAR 'RELEASE ERROR' INDICATOR
063410  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
063416  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
063424  060037  001122                  ADD     R0,$BDADR        ;REGISTER BASE ADDRESS FOR TYPEOUT
063430  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
063436  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063444  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT B
063452  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
063460  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063466  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
063474  001404                          BEQ     66$              ;BR IF STATUS FROM PORT A ZERO
063476  005737  001164                  TST     $TMP0            ;IS STATUS FROM PORT B ZERO ?
063502  001401                          BEQ     66$              ;BR IF ZERO
063504  104031                          EMT     31
063506  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT     ;CHECK STATUS FROM PORT B
063514  013737  001226  001240          MOV     PORTB,PTNBR      ;CHANGE PORT ADDRESS FOR TYPEOUT
063522  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;COMPARE WITH CONSTANT
063530  001401                          BEQ     67$              ;BR IF OK
063532  104027                          EMT     27
063534  000240                  67$:    NOP

                                ;RELEASE THE DRIVE FROM PORT B

063536  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
063544  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063552  012760  000013  000000          MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

063560  005037  001254                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
063564  012737  000012  001122          MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
063572  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
063576  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
063604  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
063612  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
063620  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
063626  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
063634  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063642  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
063650  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
063656  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
063664  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
063672  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063700  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REG'STER THE SAME FROM BOTH PORTS ?
```

```
        063706  001006                          BNE     68$             ;BR IF NOT
        063710  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
        063714  001045                          BNE     70$             ;BR IF NOT
        063716  104046                          EMT     46
        063720  000137  064104                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
        063724  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
        063732  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
        063740  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
        063746  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
        063752  001414                          BEQ     69$             ;BR IF ZERO
        063754  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
        063762  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
        063770  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
        063776  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
        064002  001012                          BNE     70$             ;BR IF NOT
        064004  012737  177777  001254  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
        064012  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
        064020  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
        064026  104026                          EMT     26
        064030  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
        064036  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
        064044  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
        064052  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
        064054  104007                          EMT     7
        064056  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
        064064  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
        064072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
        064100  001401                          BEQ     72$             ;BR IF OK
        064102  104007                          EMT     7
        064104  000240                  72$:    NOP
        064106  000004                          SCOPE                   ;LOOP ?
   1746 064110  000137  064336                  JMP     TST51           ;GO TO THE NEXT TEST

                                                ;CHECK THE REGISTERS ON THE SELECTED PORT

        064114                          TST50B:
        064114  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        064120  016037  000014  001126          MOV     RMER1(R0),$BDDAT  ;GET CONTENTS OF RMER1
        064126  012737  000014  001122          MOV     #RMER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        064134  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        064140  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
        064144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
        064152  001403                          BEQ     64$             ;BR IF OK
        064154  104006                          EMT     6
        064156  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        064162  016037  000000  001126  64$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
        064170  012737  000000  001122          MOV     #RMCS1,$BDADR   ;FORM ADDRESS OF REGISTER
        064176  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
        064202  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
        064210  001404                          BEQ     65$             ;BR IF NOT
        064212  104011                          EMT     11
        064214  012760  040000  000000          MOV     #TRE,RMCS1(R0)  ;CLEAR 'MCPE'
        064222  000240                  65$:    NOP
        064224  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        064230  016037  000042  001126          MOV     RMER2(R0),$BDDAT  ;GET CONTENTS OF RMER2
        064236  012737  000042  001122          MOV     #RMER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        064244  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
```

```
064250  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
064254  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
064262  001403                          BEQ     66$             ;BR IF OK
064264  104006                          EMT     6
064266  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
064272  016037  000000  001126  66$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
064300  012737  000000  001122          MOV     #RMCS1,$BDADR   ;FORM ADDRESS OF REGISTER
064306  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
064312  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
064320  001404                          BEQ     67$             ;BR IF NOT
064322  104011                          EMT     11
064324  012760  040000  000000          MOV     #TRE,RMCS1(R0)  ;CLEAR 'MCPE'
064332  000240                  67$:    NOP
064334  000207                          RTS     PC              ;RETURN
1747
1772
1773                            ;;*****************************************************************
                                ;*TEST 51        PORT 'B' SEIZE ACCESS TEST
                                ;*
                                ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
                                ;*
                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
                                ;*
                                ;*  B.  WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.
                                ;*
                                ;*  C.  READ RMER1, RMER2 THROUGH PORT 'A'.  VERIFY THAT PORT
                                ;*      'A' SEES 0'S FROM EACH OF THESE REGISTERS.
                                ;*
                                ;*  D.  CLEAR RMER1, RMER2 THROUGH PORT 'B'.
                                ;*
                                ;*  E.  WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.  VERIFY THAT
                                ;*      PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
                                ;*
                                ;*  F.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE HAS
                                ;*      SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
                                ;*      SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
                                ;*
                                ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
064336                          TST51:
064336  005737  001300                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
064342  001406                          BEQ     2$              ;BR IF NOT
064344  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
064346  000137  003062                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
064352  012737  177777  001300  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
064360  012737  064374  001106  2$:     MOV     #TEST51,$LPADR  ;SETUP SCOPE LOOP ADDRESS
064366  012737  064374  001110          MOV     #TEST51,$LPERR  ;SETUP ERROR LOOP ADDRESS
064374                          TEST51:
064374  112737  000051  001102          MOVB    #51,$TSTNM      ;MOVE #51 TO TEST NUMBER
064402  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
064406  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1774
1775
                                ;CLEAR ATTENTION BITS FOR BOTH PORTS
```

```
064414  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
064422  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE
064426  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
064434  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
064442  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
064450  005060  000012                  CLR     RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
064454  012760  000011  000000          MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
064462  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

064470  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
064476  013737  001226  001242          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
064504  005060  000012                  CLR     RMDS(R0)        ;WRITE RMDS
064510  013737  001224  001244          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
064516  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD 1'S INTO RMER1 THROUGH PORT B
064524  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD 1'S INTO RMER2 THROUGH PORT B
064532  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
064540  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064546  004737  065406                  JSR     PC,TST51B       ;CHECK THE REGISTERS THROUGH PORT A
064552  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
064560  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064566  005060  000042                  CLR     RMER2(R0)       ;CLEAR RMER2 ON PORT B
064572  005060  000014                  CLR     RMER1(R0)       ;CLEAR RMER1 ON PORT B
064576  013760  001236  000016          MOV     ASR1,RMAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT B
064604  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
064612  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064620  012760  177777  000014          MOV     #-1,RMER1(R0)   ,LOAD 1'S INTO RMER1 THROUGH PORT A
064626  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD 1'S INTO RMER2 THROUGH PORT A
064634  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
064642  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064650  004737  065406                  JSR     PC,TST51B       ;CHECK THE REGISTERS THROUGH PORT B

                                ;RELF^SE THE DRIVE FROM PORT B

064654  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
064662  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064670  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

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

064676  005037  001254                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
064702  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
064710  012737  000012  001122          MOV     #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
064716  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
064722  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
064730  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064736  016037  000012  001164          MOV     RMDS(R0),$TMP0  ;READ STATUS REGISTER FROM PORT A
064744  113760  001226  000010          MOVB    PORTB,RMCS2(R0)  ;SELECT PORT B
064752  013737  001226  001240          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064760  016037  000012  001126          MOV     RMDS(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
064766  001404                          BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
064770  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
064774  001401                          BEQ     66$             ;BR IF ZERO
064776  104031                          EMT     31
065000  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
065006  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
```

```
065014  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
065022  001401                          BEQ     67$             ;BR IF OK
065024  104027                          EMT     27
065026  000240                  67$:    NOP

                                ;RELEASE THE DRIVE FROM PORT A

065030  113760  001224  000010          MOVB    PORTA,RMCS2(R0)  ;SELECT PORT A
065036  013737  001224  001240          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065044  012760  000013  000000          MOV     #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

065052  005037  001254                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
065056  012737  000012  001122          MOV     #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
065064  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
065070  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
065076  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
065104  016037  000012  001170          MOV     RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
065112  042737  024001  001170          BIC     #PIP!WRL!OM,$TMP2       ;CLEAR DONT CARES
065120  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
065126  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065134  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
065142  016037  000012  001172          MOV     RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
065150  042737  024001  001172          BIC     #PIP!WRL!OM,$TMP3       ;CLEAR DONT CARES
065156  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
065164  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065172  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
065200  001006                          BNE     68$             ;BR IF NOT
065202  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
065206  001045                          BNE     70$             ;BR IF NOT
065210  104046                          EMT     46
065212  000137  065376                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
065216  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
065224  013737  001226  001240          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065232  113760  001226  000010          MOVB    PORTB,RMCS2(R0) ;SELECT PORT B.
065240  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
065244  001414                          BEQ     69$             ;BR IF ZERO
065246  013737  001224  001240          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065254  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
065262  113760  001224  000010          MOVB    PORTA,RMCS2(R0) ;SELECT PORT A.
065270  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
065274  001012                          BNE     70$             ;BR IF NOT
065276  012737  177777  001254  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
065304  012760  000011  000000          MOV     #11,RMCS1(R0)   ;CLEAR THE DRIVE
065312  012760  000013  000000          MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
065320  104026                          EMT     26
065322  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
065330  013737  001224  001240          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
065336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
065344  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
065346  104007                          EMT     7
065350  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
065356  013737  001226  001240          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
065364  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
065372  001401                          BEQ     72$             ;BR IF OK
065374  104007                          EMT     7
```

```
        065376  000240                  72$:    NOP
        065400  000004                          SCOPE                   ;LOOP ?
   1776 065402  000137  065630                  JMP     TST52           ;GO TO THE NEXT TEST

                                                ;CHECK THE REGISTERS ON THE SELECTED PORT

        065406                          TST51B:
        065406  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        065412  016037  000014  001126          MOV     RMER1(R0),$BDDAT  ;GET CONTENTS OF RMER1
        065420  012737  000014  001122          MOV     #RMER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        065426  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        065432  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
        065436  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
        065444  001403                          BEQ     64$             ;BR IF OK
        065446  104006                          EMT     6
        065450  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        065454  016037  000000  001126  64$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
        065462  012737  000000  001122          MOV     #RMCS1,$BDADR   ;FORM ADDRESS OF REGISTER
        065470  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
        065474  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
        065502  001404                          BEQ     65$             ;BR IF NOT
        065504  104011                          EMT     11
        065506  012760  040000  000000          MOV     #TRE,RMCS1(R0)  ;CLEAR 'MCPE'
        065514  000240                  65$:    NOP
        065516  005037  001250                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
        065522  016037  000042  001126          MOV     RMER2(R0),$BDDAT  ;GET CONTENTS OF RMER2
        065530  012737  000042  001122          MOV     #RMER2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        065536  060037  001122                  ADD     R0,$BDADR       ;ADD RH/RM BASE ADDRESS
        065542  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
        065546  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
        065554  001403                          BEQ     66$             ;BR IF OK
        065556  104006                          EMT     6
        065560  005137  001250                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
        065564  016037  000000  001126  66$:    MOV     RMCS1(R0),$BDDAT  ;GET THE CONTENTS OF RMCS1
        065572  012737  000000  001122          MOV     #RMCS1,$BDADR   ;FORM ADDRESS OF REGISTER
        065600  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
        065604  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
        065612  001404                          BEQ     67$             ;BR IF NOT
        065614  104011                          EMT     11
        065616  012760  040000  000000          MOV     #TRE,RMCS1(R0)  ;CLEAR 'MCPE'
        065624  000240                  67$:    NOP
        065626  000207                          RTS     PC              ;RETURN
   1777
   1778                                 ;;*******************************************************************
   1779                                 ;PUT NEWTEST HERE
   1780                                 ;;*******************************************************************
        065630  000004                  TST52:  SCOPE
```

```
 1                                  .SBTTL  END OF PASS ROUTINE

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

 065632                             $EOP:
 065632  005737  001300                     TST     KYBCTL          ;ENTERED TEST VIA KEYBOARD COMMAND ?
 065636  001402                             BEQ     .+6             ;BR IF NOT
 065640  000137  003062                     JMP     EXEC            ;RETURN TO KEYBOARD CONTROL
 065644  005037  001102                     CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 065650  005037  001176                     CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 065654  005237  001100                     INC     $PASS           ;;INCREMENT THE PASS NUMBER
 065660  042737  100000  001100             BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 065666  005327                             DEC     (PC)+           ;;LOOP?
 065670  000001                     $EOPCT: .WORD   1
 065672  003066                             BGT     $DOAGN          ;;YES
 065674  012737                             MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 065676  000001                     $ENDCT: .WORD   1
 065700  065670                             $EOPCT
 065702  104401  065710                     TYPE    ,65$            ;;TYPE ASCIZ STRING
 065706  000407                             BR      64$             ;;GET OVER THE ASCIZ
                                    ;;65$:  .ASCIZ  <12><15>/END PASS #/
 065726                             64$:
 065726  013746  001100                     MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                    ;;TYPE PASS NUMBER
 065732  104405                             TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 065734  005737  001112                     TST     $ERTTL          ;;SEE IF ANY ERRORS THIS PASS
 065740  001431                             BEQ     $GT42P          ;;BR IF NO ERRORS TO REPORT
 065742  104401  065750                     TYPE    ,67$            ;;TYPE ASCIZ STRING
 065746  000421                             BR      66$             ;;GET OVER THE ASCIZ
                                    ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
 066012                             66$:
 066012  013746  001112                     MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                    ;;TOTAL NUMBER OF ERRORS
 066016  104405                             TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 066020  005037  001112                     CLR     $ERTTL          ;;CLEAR ERROR TOTAL
 066024  104401  001207             $GT42P: TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
 066030  013700  000042             $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 066034  001405                             BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 066036  000005                             RESET                   ;;CLEAR THE WORLD
 066040  004710                     $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 066042  000240                             NOP                     ;;SAVE ROOM
 066044  000240                             NOP                     ;;FOR
 066046  000240                             NOP                     ;;ACT11
 066050                             $DOAGN:
 066050  000137                             JMP     @(PC)+          ;;RETURN
 066052  003346                     $RTNAD: .WORD   TST1AA
 066054     377     377     000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                            .EVEN
```

```
                                        .SBTTL  CLOCK SUBROUTINES

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

066060  012737  066130  000004  CKCLK:  MOV     #CKCLK1,@#ERRVEC  ;SET UP VECTOR FOR CLOCK CHECK
066066  005037  000006                  CLR     @#ERRVEC+2        ;NEW PSW
066072  005777  113114                  TST     @$LKCSR         ;CHECK FOR KW11-P
066076  013701  001216                  MOV     $LPVEC,R1       ;KW11-P VECTOR ADDRESS
066102  012721  066212                  MOV     #CLOCK,(R1)+    ;SET UP KW11-P VECTOR
066106  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
066112  012777  177777  113074          MOV     #-1,@$LKCSB     ;LOAD COUNTER BUFFER WITH 1'S
066120  012777  000135  113064          MOV     #135,@$LKCSR    ;SET CLOCK - CNT UP, 16MS, CONT INT
066126  000425                          BR      CKCLK3
066130  062706  000004          CKCLK1: ADD     #4,SP           ;RESTORE THE STACK POINTER
066134  012737  066172  000004          MOV     #CKCLK2,@#ERRVEC  ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
066142  005777  113052                  TST     @$LKS           ;LOOK FOR KW11-L
066146  013701  001222                  MOV     $LLVEC,R1       ;KW11-L VECTOR ADDRESS
066152  012721  066212                  MOV     #CLOCK,(R1)+    ;SET UP KW11-L VECTOR
066156  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
066162  012777  000100  113030          MOV     #100,@$LKS      ;SET KW11-L INTERRUPT
066170  000404                          BR      CKCLK3
066172  062706  000004          CKCLK2: ADD     #4,SP           ;RESTORE THE STACK POINTER
066176  062716  000002                  ADD     #2,(SP)         ;INCREMENT RETURN, NO CLOCK
066202  012737  000006  000004  CKCLK3: MOV     #6,@#ERRVEC     ;RESTORE THE ERROR VECTOR
066210  000207                          RTS     PC

                                        ;ROUTINE TO COUNT CLOCK TICKS

066212  062737  000021  001256  CLOCK:  ADD     #17.,TIME       ;ADD 17 MS TO ELAPSED TIME COUNTER
066220  103003                          BCC     1$              ;BRANCH IF NO OVERFLOW
066222  012737  177777  001256          MOV     #-1,TIME        ;OVERFLOW - RESTORE MAXIMUM COUNT
066230  005737  001260          1$:     TST     WATCH           ;IS WATCH ALREADY ZERO ?
066234  001406                          BEQ     2$              ;BR IF IT IS
066236  162737  000021  001260          SUB     #17.,WATCH      ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
066244  100002                          BPL     2$              ;BR IF NOT MINUS
066246  005037  001260                  CLR     WATCH           ;CLEAR WATCH DOG COUNTER
066252  000002                  2$:     RTI                     ;RETURN

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

066254  162706  000004          TOLER:  SUB     #4,SP           ;SETUP STACK
066260  016616  000004                  MOV     4(SP),(SP)      ;SAVE STACK
066264  013546                          MOV     @(R5)+,-(SP)    ;GET TIME VALUE
066266  011666  000004                  MOV     (SP),4(SP)      ;MOVE TIME VALUE
066272  011666  000006                  MOV     (SP),6(SP)      ;MOVE VALUE AGAIN
066276  006216                          ASR     (SP)            ;DIVIDE BY 2
066300  006216                          ASR     (SP)            ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
066302  061666  000004                  ADD     (SP),4(SP)      ;CALCULATE UPPER LIMIT FOR TIMEOUT
066306  162666  000004                  SUB     (SP)+,4(SP)     ;CALCULATE LOWER LIMIT FOR TIMEOUT
066312  000205                          RTS     R5              ;RETURN WITH TOLERANCES ON THE STACK
```

```
1                                       .SBTTL  SCOPE HANDLER ROUTINE

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

066314                                  $SCOPE:
066314  104407                                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
066316  004737  066650                          JSR     PC,STOP
066322  032777  040000  112610  1$:             BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
066330  001402                                  BEQ     9$              ;;NO IF SW14=0
066332  000137  066632                          JMP     $OVER           ;;JUMP OVER SCOPE ROUTINE
066336                                  9$:
                                        ;#####START OF CODE FOR THE XOR TESTER#####
066336  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
066340  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
066344  012737  066364  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
066352  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
066356  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
066362  000517                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
066364  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
066366  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
066372  000517                                  BR      $OVER           ;;LOOP ON THE PRESENT TEST
066374                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
066374  105737  001103                  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
066400  001465                                  BEQ     3$              ;;BR IF NO
066402  022737  177777  067220                  CMP     #-1,CPSAVE      ;;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED
066410  001455                                  BEQ     2003$           ;;KICK AROUND ROUTINE IF SO
066412  013746  000004                          MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
066416  012737  066434  000004                  MOV     #2000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
066424  013737  177766  067220                  MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
066432  000406                                  BR      2001$
066434  012737  177777  067220  2000$:          MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
066442  012716  066450                          MOV     #2001$,(SP)     ;;SETUP RETURN ADDRESS
066446  000002                                  RTI
066450  012637  000004          2001$:          MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

066454  022737  177777  067220  2002$:          CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
066462  001430                                  BEQ     2003$           ;;BRANCH IF SO
066464  032737  000001  067220                  BIT     #BIT00,CPSAVE   ;;SEE IF THE POWER MONITOR BIT IS ON
066472  001424                                  BEQ     2003$           ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
066474  042737  000001  177766                  BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND TO BE SET
066502  013746  001140                          MOV     SWR,-(SP)       ;;SAVE SWR ADDRESS
066506  017646  000000                          MOV     @(SP),-(SP)     ;;SAVE SWR VALUE
066512  012737  000176  001140                  MOV     #176,SWR        ;;GET SOFTWARE SWR ADDRESS
066520  011677  112414                          MOV     (SP),@SWR       ;;GET CURRENT SWR VALUE
066524  042777  001000  112406                  BIC     #BIT09,@SWR     ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
066532  104177                                  EMT     177             ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
066534  012676  000000                          MOV     (SP)+,@(SP)     ;;RESTORE SWR TO ORIGINAL VALUE
066540  012637  001140                          MOV     (SP)+,SWR       ;;RESTORE SWR ADDRESS
```

```
                                        2003$:
066544                                  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
066544  105037  001103                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAK
066550  005037  001176                          BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
066554  032777  004000  112356          3$:     BNE     1$              ;;BR IF YES
066562  001011                                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
066564  005737  001100                          BEQ     1$              ;;      INHIBIT ITERATIONS
066570  001406                                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
066572  005237  001104                          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
066576  023737  001176  001104                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
066604  002012                          1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
066606  012737  000001  001104                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
066614  013737  066646  001176          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
066622  105237  001102                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
066626  011637  001106                  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
066632  013777  001102  112302                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
066640  013716  001106                          RTI                     ;;FIXES PS
066644  000002                          $MXCNT: 5.                      ;;MAX. NUMBER OF ITERATIONS
066646  000005
                                        ;DROP PRIORITY TO ALLOW CONSOLE INTERRUPT

066650                                  STOP:
066650  012746  000140                          MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
066654  012746  066662                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
066660  000002                                  RTI                     ;;POP NEW PC AND PS
066662                                  64$:

                                        ;RAISE PRIORITY TO INHIBIT CONSOLE INTERRUPT

066662  012746  000240                          MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
066666  012746  066674                          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
066672  000002                                  RTI                     ;;POP NEW PC AND PS
066674                                  65$:
066674  000207                                  RTS     PC              ;RETURN
```

```
1                                       .SBTTL  ERROR HANDLER ROUTINE

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

066676  105037  067222          $ERROR: CLRB    IBSAVE          ;;CLEAR THE ITEM BYTE SAVE LOCATION
066702  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
066704  113737  001102  001246          MOVB    $TSTNM,TSTNUM
066712  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
066716  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
066720  013777  001102  112214          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
066726  032777  002000  112204          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
066734  001402                          BEQ     1$              ;;NO - SKIP
066736  104401  001202                  TYPE    ,$BELL          ;;RING BELL
066742  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
066746  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
066752  162737  000002  001116          SUB     #2,$ERRPC
066760  117737  112132  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
066766  032777  001000  112144          BIT     #BIT09,@SWR     ;;SEE IF LOOP ON ERROR IS SET
066774  001060                          BNE     1004$           ;;BRANCH AROUND ROUTINE IF SO
066776  122737  000177  001114          CMPB    #177,$ITEMB     ;;SEE IF THIS IS THE POWER FAIL CALL
067004  001454                          BEQ     1004$           ;;BRANCH AROUND ROUTINE IF IT IS
067006  105737  067222                  TSTB    IBSAVE          ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
067012  001047                          BNE     1003$           ;;BRANCH IF SO
067014  022737  177777  067220          CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
067022  001445                          BEQ     1004$           ;;BRANCH IF SO
067024  013746  000004                  MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
067030  012737  067046  000004          MOV     #1000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
067036  013737  177766  067220          MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
067044  000406                          BR      1001$
067046  012737  177777  067220  1000$: MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
067054  012716  067062                  MOV     #1001$,(SP)     ;;SETUP RETURN ADDRESS
067060  000002                          RTI
067062  012637  000004          1001$: MOV      (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

067066  022737  177777  067220  1002$: CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
067074  001420                          BEQ     1004$           ;;BRANCH IF SO
067076  032737  000001  067220          BIT     #BIT00,CPSAVE   ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
067104  001414                          BEQ     1004$           ;;BRANCH IF OK
067106  042737  000001  177766          BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND SET
067114  113737  001114  067222          MOVB    $ITEMB,IBSAVE   ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
067122  112737  000177  001114          MOVB    #177,$ITEMB     ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
067130  000402                          BR      1004$           ;;BRANCH OVER IBSAVE CLEARING

067132  105037  067222          1003$: CLRB     IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
067136                          1004$:
067136  032777  020000  111774          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
067144  001004                          BNE     20$             ;;SKIP TYPEOUTS
067146  004737  067224                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
```

```
067152  104401  001207                  TYPE    ,$CRLF
067156                          20$:
067156  105737  067222          2$:     TSTB    IBSAVE          ;;SEE IF IBSAVE IS LOADED
067162  001005                          BNE     3$              ;;BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
067164  005777  111750                  TST     @SWR            ;;HALT ON ERROR
067170  100002                          BPL     3$              ;;SKIP IF CONTINUE
067172  000000                          HALT                    ;;HALT ON ERROR!
067174  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
067176                          3$:
067176  022737  066040  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
067204  001001                          BNE     6$              ;;BRANCH IF NO
067206  000000                          HALT                    ;;YES
067210                          6$:
067210  105737  067222                  TSTB    IBSAVE          ;;SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
067214  001236                          BNE     7$              ;;BRANCH BACK TO CALL ORIGINAL ERROR
067216  000002                          RTI                     ;;RETURN
067220  000000                  CPSAVE: .WORD   0               ;;LOCATION TO SAVE CPU ERROR REG CONTENTS
067222  000000                  IBSAVE: .WORD   0               ;;LOCATION TO SAVE ITEM BYTE
```

```
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

067224                          $ERRTYP:
067224  104401  001207                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
067230  010046                          MOV     R0,-(SP)        ;;SAVE R0
067232  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
067234  153700  001114                  BISB    @#$ITEMB,R0
067240  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
067242  013746  001116                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
067246  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
067250  000456                          BR      10$             ;;GET OUT
067252  122700  000177          1$:     CMPB    #177,R0         ;;SEE IF THIS ERROR CALL IS SPECIAL POWER FAIL CALL
067256  001006                          BNE     1000$           ;;BRANCH IF NOT
067260  113737  001102  067562          MOVB    $TSTNM,PFTSTN   ;;GET TEST NUMBER
067266  012700  067422                  MOV     #PFECH,R0       ;;MOVE POWER FAIL ERROR CALL TABLE TO R0
067272  000406                          BR      1001$           ;;BRANCH TO CALL ERROR
067274  005300                  1000$:  DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
067276  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
067300  006300                          ASL     R0
067302  006300                          ASL     R0
067304  062700  001310                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
067310  012037  067320          1001$:  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
067314  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
067316  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
067320  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
067322  104401  001207                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
067326  012037  067336          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
067332  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
067334  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
067336  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
067340  104401  001207                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
067344  010146                  5$:     MOV     R1,-(SP)        ;;SAVE R1
067346  012001                          MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
067350  001415                          BEQ     9$              ;;BR IF NO DATA TO BE TYPED
067352  012000                          MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
067354  105720                  6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
067356  001003                          BNE     7$              ;;BR IF DECIMAL
067360  013146                          MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
067362  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
067364  000402                          BR      8$
067366                          7$:
067366  013146                          MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
067370  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
067372  005711                  8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
067374  001403                          BEQ     9$              ;;BR IF NO
067376  104401  067416                  TYPE    ,11$            ;;TYPE TWO(2) SPACES
067402  000764                          BR      6$              ;;LOOP

067404  012601                  9$:     MOV     (SP)+,R1        ;;RESTORE R1
067406  012600                  10$:    MOV     (SP)+,R0        ;;RESTORE R0
```

```
067410  104401  001207                    TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
067414  000207                            RTS     PC              ;;RETURN
067416     040     040     000  11$:      .ASCIZ  /  /            ;;TWO(2) SPACES
                                          .EVEN
067422  067432  067514  067546  PFECH:    PFECH1,PFECH2,PFECH3,PFECH4 ;;WORDS DEFINING TABLES BELOW
067432     120     117     127  PFECH1:   .ASCIZ  ?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?
067514     124     105     123  PFECH2:   .ASCIZ  ?TESTNO  ERR PC  CPUERREG?
                                          .EVEN
067546  067562  001116  067220  PFECH3:   .WORD   PFTSTN,$ERRPC,CPSAVE,0
067556     000     000     000  PFECH4:   .BYTE   0,0,0,0
067562  000000                   PFTSTN:  .WORD   0               ;;CONTAINS TEST NUMBER FOR PF BIT ERROR
```

```
                                .SBTTL  TYPE ROUTINE

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

067564  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
067570  100002                          BPL     1$              ;;BR IF YES
067572  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
067574  000407                          BR      3$              ;;LEAVE
067576  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
067600  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
067604  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
067606  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
067610  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
067612  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
067614  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
067620  000002                          RTI                     ;;RETURN
067622  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
067626  001430                          BEQ     8$
067630  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
067634  001006                          BNE     5$
067636  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
067640  104401                          TYPE                    ;;TYPE A CR AND LF
067642  001207                          $CRLF
067644  105037  070052                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
067650  000755                          BR      2$              ;;GET NEXT CHARACTER
067652  004737  067734          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
067656  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
067662  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
067664  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
067670  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
067674  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
067676  004737  067734                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
067702  105337  070052                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
067706  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

067710  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
067714  004737  067734          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
067720  132737  000007  070052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
067726  001372                          BNE     9$              ;;TAB STOP
067730  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
067732  000724                          BR      2$              ;;GET NEXT CHARACTER
```

```
067734                          $TYPEC:
067734  105777  111204                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
067740  100022                          BPL     10$             ;;BR IF NOT
067742  017746  111200                  MOV     @$TKB,-(SP)     ;;GET CHAR
067746  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
067752  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
067756  001012                          BNE     102$            ;;BR IF NOT
067760                          101$:
067760  105777  111160                  TSTB    @$TKS           ;;WAIT FOR CHAR
067764  100375                          BPL     101$
067766  117716  111154                  MOVB    @$TKB,(SP)      ;;GET CHAR
067772  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
067776  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
070002  001366                          BNE     101$            ;;BR IF NOT
070004                          102$:
070004  005726                          TST     (SP)+           ;;FIX STACK
070006                          10$:
070006  105777  111136                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
070012  100375                          BPL     10$
070014  116677  000002  111130          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
070022  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
070030  001003                          BNE     1$              ;;BRANCH IF NO
070032  105037  070052                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
070036  000406                          BR      $TYPEX          ;;EXIT
070040  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
070046  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
070050  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
070052  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
070054  000207                  $TYPEX: RTS     PC
```

```
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

070056  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
070062  116637  000001  070301          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
070070  112637  070303                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
070074  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
070100  000406                          BR      $TYPON
070102  112737  000001  070301  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
070110  112737  000006  070303          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
070116  112737  000005  070300  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
070124  010346                          MOV     R3,-(SP)        ;;SAVE R3
070126  010446                          MOV     R4,-(SP)        ;;SAVE R4
070130  010546                          MOV     R5,-(SP)        ;;SAVE R5
070132  113704  070303                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
070136  005404                          NEG     R4
070140  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
070144  110437  070302                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
070150  113704  070301                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
070154  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
070160  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
070162  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
070164  000404                          BR      3$              ;;GO DO MSB
070166  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
070170  006105                          ROL     R5
070172  006105                          ROL     R5
070174  010503                          MOV     R5,R3
070176  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
070200  105337  070302                  DECB    $OMODE          ;;TYPE THIS DIGIT?
070204  100016                          BPL     7$              ;;BR IF NO
070206  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
070212  001002                          BNE     4$              ;;TEST FOR 0
070214  005704                          TST     R4              ;;SUPPRESS THIS 0?
070216  001403                          BEQ     5$              ;;BR IF YES
070220  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
```

```
070222  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
070226  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
070232  110337  070276                  MOVB    R3,8$           ;;SAVE FOR TYPING
070236  104401  070276                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
070242  105337  070300          7$:     DECB    $OCNT           ;;COUNT BY 1
070246  003347                          BGT     2$              ;;BR IF MORE TO DO
070250  002402                          BLT     6$              ;;BR IF DONE
070252  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
070254  000744                          BR      2$              ;;GO DO THE LAST DIGIT
070256  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
070260  012604                          MOV     (SP)+,R4        ;;RESTORE R4
070262  012603                          MOV     (SP)+,R3        ;;RESTORE R3
070264  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
070272  012616                          MOV     (SP)+,(SP)
070274  000002                          RTI                     ;;RETURN
070276     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
070277     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
070300     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
070301     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
070302  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;* UMBER 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

070304                          $TYPDS:
070304  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
070306  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
070310  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
070312  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
070314  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
070316  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
070322  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
070326  100004                          BPL     1$              ;;BR IF INPUT IS POS.
070330  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
070332  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
070340  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
070342  012703  070520                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
070346  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
070352  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
070354  016001  070510                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
070360  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
070362  002402                          BLT     4$              ;;BR IF DONE
070364  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
070366  000774                          BR      3$
070370  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
070372  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
070374  001002                          BNE     5$              ;;FALL THROUGH IF 0
070376  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
070400  100407                          BMI     7$              ;;BR IF YES
070402  106316                  5$:     ASLB    (SP)            ;;MSD?
070404  103003                          BCC     6$              ;;BR IF NO
070406  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
070414  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
070420  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
070424  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
070426  005720                          TST     (R0)+           ;;JUST INCREMENTING
070430  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
070434  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
070436  003002                          BGT     8$              ;;GO TO EXIT
070440  010502                          MOV     R5,R2           ;;GET THE LSD
070442  000764                          BR      6$              ;;GO CHANGE TO ASCII
070444  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
070446  100003                          BPL     9$              ;;BR IF NO
070450  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
070456  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
070460  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
070462  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
070464  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
070466  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
070470  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
070472  104401  070520                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
070476  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
070504  012616                          MOV     (SP)+,(SP)
070506  000002                          RTI                     ;;RETURN TO USER
070510  023420                  $DTBL:  10000.
070512  001750                          1000.
070514  000144                          100.
070516  000012                          10.
070520                          $DBLK:  .BLKW   4
```

```
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*****************************************************************
                                .ENABL  LSB
070530  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
070532  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
070534  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
070536                          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        070537                  $TKQEND=.
                                .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
070540  005037  070530          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
070544  012737  070536  070532          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
070552  013737  070532  070534          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
070560  012737  070610  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
070566  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
070574  005777  110346                  TST     @$TKB           ;;CLEAR DONE FLAG
070600  012777  000100  110336          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
070606  000207                          RTS     PC              ;;RETURN TO CALLER

                                ;*TK SERVICE ROUTINE
                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                ;*IT IN THE QUEUE.
                                ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
                                ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START)
                                ;
070610  117746  110332          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
070614  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
070620  021627  000021                  CMP     (SP),#$XON      ;;IS IT A RANDOM XON?
070624  001002                          BNE     30$             ;;BRANCH IF NO
070626  005726                          TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK
070630  000002                          RTI                     ;;RETURN
070632                          30$:
070632  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
070636  001007                          BNE     1$              ;;BRANCH IF NO
070640  104401  071736                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
070644  004737  070540                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
070650  005726                          TST     (SP)+           ;;CLEAN UP STACK
070652  000137  002240                  JMP     START           ;;CONTROL C RESTART
070656  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
070662  001004                          BNE     2$              ;;BRANCH IF NO
070664  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
070672  001500                          BEQ     6$              ;;GO TO SWR CHANGE

070674                          2$:
070674  022737  000001  070530          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
070702  001004                          BNE     3$              ;;BRANCH IF NO
070704  104401  001202                  TYPE    ,$BELL          ;;RING THE TTY BELL
```

```
070710  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
070712  000451                          BR      5$              ;;EXIT
070714  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
070720  001021                          BNE     32$             ;;BRANCH IF NO
070722  005077  110216                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
070726  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
070730  105777  110210          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
070734  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
070736  117746  110204                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
070742  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
070746  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
070752  001366                          BNE     31$             ;;BRANCH IF NO
070754  012777  000100  110162          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
070762  000002                          RTI                     ;;RETURN
070764  005237  070530          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
070770  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
070774  002405                          BLT     4$              ;;BRANCH IF YES
070776  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
071002  003002                          BGT     4$              ;;BRANCH IF YES
071004  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
071010  112677  177516          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
071014  005237  070532                  INC     $TKQIN          ;;UPDATE THE POINTER
071020  023727  070532  070537          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
071026  001003                          BNE     5$              ;;BRANCH IF NO
071030  012737  070536  070532          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
071036  000002                  5$:     RTI                     ;;RETURN

                                ;;*****************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
071040  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
071046  001124                          BNE     15$             ;;EXIT IF NOT
071050  105777  110070                  TSTB    @$TKS           ;;IS A CHAR WAITING?
071054  100121                          BPL     15$             ;;IF NOT, EXIT
071056  117746  110064                  MOVB    @$TKB,-(SP)     ;;YES
071062  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
071066  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
071072  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                                ;;AND EXIT

                                ;;*****************************************************************
                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
071074  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
071102  001674                          BEQ     2$              ;;BRANCH IF YES
071104  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
071106  004737  070540                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
071112  005077  110026                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
071116  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

071124  104401  071750                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
071130  104401  071755          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
071134  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
071140  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
071142  104401  071766                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
071146  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
071150  005046                          CLR     -(SP)           ;;THE NEW SWR
071152  105777  107766          7$:     TSTB    @$TKS           ;;CHAR THERE?
071156  100375                          BPL     7$              ;;IF NOT TRY AGAIN

071160  117746  107762                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
071164  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

071170  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
071174  001015                          BNE     9$              ;;BRANCH IF NOT
071176  104401  071736                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
071202  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
071206  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
071214  001003                          BNE     8$              ;;BRANCH IF NO
071216  012777  000100  107720          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
071224  000137  002240          8$:     JMP     START           ;;CONTROL-C RESTART


071230  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
071234  001005                          BNE     10$             ;;BRANCH IF NOT
071236  104401  071743                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
071242  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
071246  000737                          BR      19$             ;;LET'S TRY IT AGAIN


071250  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
071254  001022                          BNE     16$             ;;BRANCH IF NO
071256  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
071262  001403                          BEQ     11$             ;;BRANCH IF YES
071264  016677  000002  107646          MOV     2(SP),@SWR      ;;SAVE NEW SWR
071272  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
071276  104401  001207          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
071302  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
071310  001003                          BNE     15$             ;;BRANCH IF NOT
071312  012777  000100  107624          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
071320  000002                  15$:    RTI                     ;;RETURN
071322  004737  067734          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
071326  021627  000060                  CMP     (SP),#60        ;;CHAR < 0'
071332  002420                          BLT     18$             ;;BRANCH IF YES
071334  021627  000067                  CMP     (SP),#67        ;;CHAR > 7'
071340  003015                          BGT     18$             ;;BRANCH IF YES
071342  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
071346  005766  000002                  TST     -(SP)           ;;IS THIS THE FIRST CHAR
071352  001403                          BEQ     17$             ;;BRANCH IF YES
071354  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
071356  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
071360  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
071362  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
071366  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
071372  000667                          BR      7$              ;;GET THE NEXT ONE
071374  104401  001206          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
071400  000720                          BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB

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

```
                                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*CALL:
                                        ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                        ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                        ;

071402  011646                          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
071404  016666  000004  000002                  MOV     4(SP),2(SP)     ;;THE PS
071412  005066  000004                          CLR     4(SP)           ;;GET READY FOR A CHARACTER
071416  005046                                  CLR     -(SP)           ;;PUT NEW PS ON STACK
071420  012746  071426                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
071424  000002                                  RTI                     ;;POP NEW PC AND PS
071426                                  64$:
071426  005737  070530                  1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
071432  001775                                  BEQ     1$
071434  005337  070530                          DEC     $TKCNT          ;;DECREMENT THE COUNTER
071440  117766  177070  000004                  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
071446  005237  070534                          INC     $TKQOUT         ;;UPDATE THE POINTER
071452  023727  070534  070537                  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
071460  001003                                  BNE     2$              ;;BRANCH IF NO
071462  012737  070536  070534                  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
071470  000002                          2$:     RTI                     ;;RETURN
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        ;*CALL:
                                        ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

071472  010346                          $RDLIN: MOV     R3, (SP)        ;;SAVE R3
071474  005046                                  CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
071476  012703  071726                  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
071502  022703  071736                  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
071506  101456                                  BLOS    4$              ;;BR IF YES
071510  104410                                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
071512  112613                                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
071514  122713  000177                  10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
071520  001022                                  BNE     5$              ;;BR IF NO
071522  005716                                  TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
071524  001007                                  BNE     6$              ;;BR IF NO
071526  112737  000134  071724                  MOVB    #'\,9$          ;;TYPE A BACK SLASH
071534  104401  071724                          TYPE    ,9$
071540  012716  177777                          MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
071544  005303                          6$:     DEC     R3              ;;BACKUP BY ONE
  1546  020327  071726                          CMP     R3,#$TTYIN      ;;STACK EMPTY?
  1552  103434                                  BLO     4$              ;;BR IF YES
071554  111337  071724                          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
071560  104401  071724                          TYPE    ,9$             ;;GO TYPE
071564  000746                                  BR      2$              ;;GO READ ANOTHER CHAR.
071566  005716                          5$:     TST     (SP)            ;;RUBOUT KEY SET?
071570  001406                                  BEQ     7$              ;;BR IF NO
071572  112737  000134  071724                  MOVB    #'\,9$          ;;TYPE A BACK SLASH
071600  104401  071724                          TYPE    ,9$
071604  005016                                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
071606  122713  000025                  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
071612  001003                                  BNE     8$              ;;BR IF NO
```

```
071614  104401  071743                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U'
071620  000726                          BR      1$              ;;GO START OVER
071622  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A '^R'?
071626  001011                          BNE     3$              ;;BRANCH IF NO
071630  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
071632  104401  001207                  TYPE    ,$CRLF          ;;TYPE A 'CR' & 'LF'
071636  104401  071726                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
071642  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
071644  104401  001206          4$:     TYPE    ,$QUES          ;;TYPE A '?'
071650  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
071652  111337  071724          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
071656  104401  071724                  TYPE    ,9$
071662  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
071666  001305                          BNE     2$              ;;LOOP IF NOT RETURN
071670  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
071674  104401  001210                  TYPE    ,$LF            ;;TYPE A LINE FEED
071700  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
071702  012603                          MOV     (SP)+,R3        ;;RESTORE R3
071704  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
071706  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
071714  012766  071726  000004          MOV     #$TTYIN,4(SP)
071722  000002                          RTI                     ;;RETURN
071724     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
071725     000                          .BYTE   0               ;;TERMINATOR
071726                          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
071736     136     103     015  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL 'C'
071743     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
071750     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
071755     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
071766     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
                                .EVEN
```

```
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

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

072000  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
072002  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
072010  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
072012  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
072014  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
072016  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
072020  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
072022  010037  072126                  MOV     R0,5$           ;;AND SAVE IT
072026  005001                          CLR     R1              ;;CLEAR DATA WORD
072030  005002                          CLR     R2
072032  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
072034  001420                          BEQ     3$              ;;IF ZERO GET OUT
072036  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
072042  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
072044  122716  000067                  CMPB    #'7,(SP)
072050  002423                          BLT     4$
072052  006301                          ASL     R1              ;;*2
072054  006102                          ROL     R2
072056  006301                          ASL     R1              ;;*4
072060  006102                          ROL     R2
072062  006301                          ASL     R1              ;;*8
072064  006102                          ROL     R2
072066  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
072072  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
072074  000756                          BR      2$              ;;LOOP
072076  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
072100  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
072104  010237  072136                  MOV     R2,$HIOCT
072110  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
072112  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
072114  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
072116  000002                          RTI                     ;;RETURN
072120  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
072122  105010                          CLRB    (R0)            ;;SET A TERMINATOR
072124  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
072126  000000                  5$:     .WORD   0
072130  104401  001206                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
072134  000730                          BR      1$              ;;TRY AGAIN
072136  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

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

072140                          $SAVREG:
072140  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
072142  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
072144  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
072146  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
072150  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
072152  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
072154  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
072160  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
072164  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
072170  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
072174  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
072176                          $RESREG:
072176  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
072202  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
072206  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
072212  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
072216  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
072220  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
072222  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
072224  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
072226  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
072230  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
072232  000002                          RTI
```

```
                                .SBTTL  TRAP DECODER

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

072234  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
072236  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
072242  005740                          TST     -(R0)           ;;BACKUP BY 2
072244  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
072246  006300                          ASL     R0              ;;POSITION FOR INDEXING
072250  016000  072270                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
072254  000200                          RTS     R0              ;;GO TO ROUTINE


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

072256  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
072260  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
072266  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
072270  072256                  $TRPAD: .WORD   $TRAP2
072272  067564                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
072274  070102                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
072276  070056                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
072300  070116                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
072302  070304                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

072304  071130                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

072306  071040                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
072310  071402                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
072312  071472                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
072314  072000                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
072316  072140                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
072320  072176                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
```

```
                                        .SBTTL  TELETYPE MESSAGES
 2
 3 072322     200     105     116  ENTERA: .ASCIZ  <CRLF>/ENTER DRIVE ADDRESS: /
 4 072351     040     077     111  ADRERR: .ASCIZ  / ?INVALID ADDRESS/<CRLF>
 5 072374     200     120     117  PORTAIS: .ASCIZ <CRLF>/PORT 'A' ADDRESS IS: /
 6 072423     200     120     117  PORTBIS: .ASCIZ <CRLF>/PORT 'B' ADDRESS IS: /
 7 072452     200     116     117  NOCLOCK: .ASCIZ <CRLF>/NO SYSTEM 'L' OR 'P' CLOCK/<CRLF><LF>
 8 072510     012     105     116  TESTNO: .ASCIZ  <LF>/ENTER TEST #: /
 9 072530     040     077     111  BADNO:  .ASCIZ  / ?INVALID TEST NUMBER/<CRLF>
10 072557     040     105     122  TSTERR: .ASCIZ  / ERRORS/<CRLF>
11 072570     200     012     122  ADDRIS: .ASCIZ  <CRLF><LF>@RH/RM ADDRESS (RMCS1) IS: @
12 072625     012     105     116  NTRH:   .ASCIZ  <LF>@ENTER RH/RM ADDRESS: @
```

```
                                .SBTTL  TEST ERROR MESSAGES

072654    127    122    117  EM1:    .ASCIZ  /WRONG DRIVE TYPE/
072675    104    122    111  EM2:    .ASCIZ  /DRIVE NOT ON LINE/
072717    123    105    122  EM3:    .ASCIZ  /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
073001    104    122    111  EM4:    .ASCIZ  /DRIVE NOT SEIZED BY PORT/
073032    127    122    117  EM5:    .ASCIZ  /WRONG STATUS SEEN BY THE SEIZING PORT/
073100    122    105    107  EM6:    .ASCIZ  /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
073200    122    105    107  EM7:    .ASCIZ  /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
073261    122    105    107  EM10:   .ASCIZ  /REGISTER CONTENTS WRONG/
073311    103    117    116  EM11:   .ASCIZ  /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
073375    104    122    111  EM12:   .ASCIZ  /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
073445    122    105    101  EM13:   .ASCIZ  /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
073532    126    117    114  EM14:   .ASCIZ  /VOLUME VALID SET ON THE WRONG PORT/
073575    101    124    124  EM15:   .ASCIZ  /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
073654    101    124    124  EM16:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
073727    101    124    124  EM17:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
074006    104    122    111  EM20:   .ASCIZ  /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
074066    104    122    111  EM21:   .ASCIZ  /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
074141    104    122    111  EM22:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
074226    124    111    115  EM23:   .ASCIZ  /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
074274    122    105    114  EM24:   .ASCIZ  /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
074353    124    111    115  EM25:   .ASCIZ  /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
074416    104    122    111  EM26:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
074503    122    105    107  EM27:   .ASCIZ  /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
074561    104    122    111  EM30:   .ASCIZ  /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
074656    104    122    111  EM31:   .ASCIZ  /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
074733    101    124    124  EM32:   .ASCIZ  /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
075004    104    122    111  EM33:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
075106    104    122    111  EM34:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
075211    124    111    115  EM35:   .ASCIZ  /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
075270    124    111    115  EM36:   .ASCIZ  /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
075342    104    122    111  EM37:   .ASCIZ  /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
075410    101    124    124  EM40:   .ASCIZ  /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
075465    124    111    115  EM41:   .ASCIZ  /TIMEOUT CLEARED THE ATTENTION BIT/
075527    104    122    111  EM42:   .ASCIZ  /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
075615    104    122    111  EM43:   .ASCIZ  /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
075672    127    122    111  EM44:   .ASCIZ  /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
075747    120    117    122  EM45:   .ASCIZ  @PORT SELECT SWITCH ON DRIVE, NOT IN 'A/B'@
076021    103    101    116  EM46:   .ASCIZ  /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
076070    101    124    124  EM47:   .ASCIZ  /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
076156    101    124    124  EM50:   .ASCIZ  /ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR/
076240    101    124    124  EM51:   .ASCIZ  /ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
076327    124    110    105  EM52:   .ASCIZ  /THE ATTN BIT SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
076422    103    101    116  EM53:   .ASCIZ  /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
076503    122    105    114  EM54:   .ASCIZ  /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
076576    124    111    115  EM55:   .ASCIZ  /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
076643    122    110    057  EM56:   .ASCIZ  @RH/RM DIDN'T RESPOND TO ADDRESSING@
076706    120    117    122  EM57:   .ASCIZ  /PORT REQUEST FLOP(S) WRONG STATE/
076747    101    124    124  EM60:   .ASCIZ  /ATTENTION NOT RESET BY WRITING RMAS/
077013    101    124    124  EM61:   .ASCIZ  /ATTENTION NOT RESET BY GO/
077045    101    124    124  EM62:   .ASCIZ  /ATTENTION RESET BY GO WHEN NOT SEIZED/
077113    104    122    111  EM63:   .ASCIZ  /DRIVE SEIZED BY 'MUR' CHANGE IN RMMR1/
077161    101    124    124  EM64:   .ASCIZ  /ATTENTION NOT SET BY 'MUR' CHANGE IN RMMR1/
077234    126    117    114  EM65:   .ASCIZ  /VOLUME VALID NOT RESET BY 'MUR' IN RMMR1 /
```

```
077306    124    105    123  DH1:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
077357    124    105    123  DH3:    .ASCIZ  /TEST #  ERR PC  REG ADR PORT A  PORT B/
077426    040    040    040  DH4:    .ASCII  /                SEIZE   ERROR/<CR><LF>
077465    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
077551    124    105    123  DH5:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR GOOD    BAD/
077625    040    040    040  DH7:    .ASCII  /                RELSNG  ERROR/<CR><LF>
077664    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
077750    124    105    123  DH11:   .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
100021    040    040    040  DH13:   .ASCII  /                SEIZE   ERROR/<CR><LF>
100060    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR CONTENTS/
100141    040    040    040  DH22:   .ASCII  /                RELSNG  SEIZE/<CR><LF>
100200    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
100237    040    040    040  DH23:   .ASCII  /                SEIZE/<CR><LF>
100266    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
100337    040    040    040  DH26:   .ASCII  /                RELSNG/<CR><LF>
100367    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #/
100416    040    040    040  DH31:   .ASCII  /                RELSNG  RQSTNG/<CR><LF>
100456    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
100515    124    105    123  DH36:   .ASCIZ  /TEST #  ERR PC  PORT #/
100544    124    105    123  DH42:   .ASCIZ  /TEST #  ERR PC/
100563    040    040    040  DH44:   .ASCII  /                RELSNG  ERROR/<CR><LF>
100622    124    105    123          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
100661    040    040    040  DH46:   .ASCII  /                PORT A  PORT B/<CR><LF>
100721    124    105    123          .ASCIZ  /TEST #  ERR PC  RMDS    RMDS/
100756    124    105    123  DH55:   .ASCIZ  /TEST #  ERR PC  PORT #  TIMEOUT VALUE (IN MS)/
101034    044    122    115  DH56:   .ASCIZ  /$RMADR/
101043    124    105    123  DH57:   .ASCII  /TEST #  ERR PC      PORT A          PORT B/
101115    040    040    040          .ASCIZ  /                EXPCTD  RECEVD  EXPECTD  RECEVD/

                               .EVEN

101176  001246  001116  001240  DT1:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$BDDAT,0
101212  001246  001116  001122  DT3:    .WORD   TSTNUM,$ERRPC,$BDADR,$GDDAT,$BDDAT,0
101226  001246  001116  001240  DT5:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$GDDAT,$BDDAT,0
101244  001246  001116  001244  DT6:    .WORD   TSTNUM,$ERRPC,OPPRT,$BDADR,$BDDAT,0
101260  001246  001116  001242  DT7:    .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$GDDAT,$BDDAT,0
101300  001246  001116  001242  DT13:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$BDDAT,0
101316  001246  001116  001242  DT22:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,0
101330  001246  001116  001242  DT23:   .WORD   TSTNUM,$ERRPC,SEIZPT,$BDADR,$BDDAT,0
101344  001246  001116  001242  DT31:   .WORD   TSTNUM,$ERRPC,SEIZPT,OPPRT,0
101356  001246  001116  001242  DT36:   .WORD   TSTNUM,$ERRPC,SEIZPT,0
101366  001246  001116  001240  DT37:   .WORD   TSTNUM,$ERRPC,PTNBR,0
101376  001246  001116  000000  DT42:   .WORD   TSTNUM,$ERRPC,0
101404  001246  001116  001170  DT46:   .WORD   TSTNUM,$ERRPC,$TMP2,$TMP3,0
101416  001246  001116  001244  DT54:   .WORD   TSTNUM,$ERRPC,OPPRT,SEIZPT,0
101430  001246  001116  001242  DT55:   .WORD   TSTNUM,$ERRPC,SEIZPT,TIME,0
101442  001304  000000          DT56:   .WORD   $RMADR,0
101446  001246  001116  001164  DT57:   .WORD   TSTNUM,$ERRPC,$TMP0,$TMP1,$TMP2,$TMP3,0

101464     000     000     000  DF1:    .BYTE   0,0,0,0,0
101471     000     000     000  DF5:    .BYTE   0,0,0,0,0,0
101477     000     000     000  DF7:    .BYTE   0,0,0,0,0,0,0
101506     000     000     000  DF31:   .BYTE   0,0,0,0
101512     000     000     000  DF36:   .BYTE   0,0,0
101515     000     000          DF42:   .BYTE   0,0
101517     000     000     000  DF55:   .BYTE   0,0,0,1
101523     000                  DF56:   .BYTE   0
```

```
58                                  .EVEN
```

```
                                .SBTTL  CONSTANTS, TABLES, ETC

                                ;TABLE OF TEST STARTING ADDRESSES

101524  003364                  TSTADR: .WORD   TST1            ;STARTING ADDRESS OF TEST 1
101526  004676                          .WORD   TST2            ;STARTING ADDRESS OF TEST 2
101530  006244                          .WORD   TST3            ;STARTING ADDRESS OF TEST 3
101532  007612                          .WORD   TST4            ;STARTING ADDRESS OF TEST 4
101534  010742                          .WORD   TST5            ;STARTING ADDRESS OF TEST 5
101536  012072                          .WORD   TST6            ;STARTING ADDRESS OF TEST 6
101540  012544                          .WORD   TST7            ;STARTING ADDRESS OF TEST 7
101542  013216                          .WORD   TST10           ;STARTING ADDRESS OF TEST 10
101544  014462                          .WORD   TST11           ;STARTING ADDRESS OF TEST 11
101546  015726                          .WORD   TST12           ;STARTING ADDRESS OF TEST 12
101550  017046                          .WORD   TST13           ;STARTING ADDRESS OF TEST 13
101552  020166                          .WORD   TST14           ;STARTING ADDRESS OF TEST 14
101554  021566                          .WORD   TST15           ;STARTING ADDRESS OF TEST 15
101556  023166                          .WORD   TST16           ;STARTING ADDRESS OF TEST 16
101560  024112                          .WORD   TST17           ;STARTING ADDRESS OF TEST 17
101562  025036                          .WORD   TST20           ;STARTING ADDRESS OF TEST 20
101564  026104                          .WORD   TST21           ;STARTING ADDRESS OF TEST 21
101566  027152                          .WORD   TST22           ;STARTING ADDRESS OF TEST 22
101570  031226                          .WORD   TST23           ;STARTING ADDRESS OF TEST 23
101572  031752                          .WORD   TST24           ;STARTING ADDRESS OF TEST 24
101574  033146                          .WORD   TST25           ;STARTING ADDRESS OF TEST 25
101576  034342                          .WORD   TST26           ;STARTING ADDRESS OF TEST 26
101600  036036                          .WORD   TST27           ;STARTING ADDRESS OF TEST 27
101602  037532                          .WORD   TST30           ;STARTING ADDRESS OF TEST 30
101604  041226                          .WORD   TST31           ;STARTING ADDRESS OF TEST 31
101606  042722                          .WORD   TST32           ;STARTING ADDRESS OF TEST 32
101610  044200                          .WORD   TST33           ;STARTING ADDRESS OF TEST 33
101612  045324                          .WORD   TST34           ;STARTING ADDRESS OF TEST 34
101614  046222                          .WORD   TST35           ;STARTING ADDRESS OF TEST 35
101616  047120                          .WORD   TST36           ;STARTING ADDRESS OF TEST 36
101620  050116                          .WORD   TST37           ;STARTING ADDRESS OF TEST 37
101622  051114                          .WORD   TST40           ;STARTING ADDRESS OF TEST 40
101624  053100                          .WORD   TST41           ;STARTING ADDRESS OF TEST 41
101626  055064                          .WORD   TST42           ;STARTING ADDRESS OF TEST 42
101630  056340                          .WORD   TST43           ;STARTING ADDRESS OF TEST 43
101632  057614                          .WORD   TST44           ;STARTING ADDRESS OF TEST 44
101634  060444                          .WORD   TST45           ;STARTING ADDRESS OF TEST 45
101636  061274                          .WORD   TST46           ;STARTING ADDRESS OF TEST 46

                                ;ATTENTION BIT TABLE

101640     001                  ATABIT: .BYTE   1               ;ATTENTION BIT FOR DRIVE 0
101641     002                          .BYTE   2               ;ATTENTION BIT FOR DRIVE 1
101642     004                          .BYTE   4               ;ATTENTION BIT FOR DRIVE 2
101643     010                          .BYTE   10              ;ATTENTION BIT FOR DRIVE 3
101644     020                          .BYTE   20              ;ATTENTION BIT FOR DRIVE 4
101645     040                          .BYTE   40              ;ATTENTION BIT FOR DRIVE 5
101646     100                          .BYTE   100             ;ATTENTION BIT FOR DRIVE 6
101647     200                          .BYTE   200             ;ATTENTION BIT FOR DRIVE 7

101650  000052                  MAXTN:  .WORD   52              ;MAXIMUM TEST NUMBER

        000200                  .END    200
```

SYMBOL TABLE

```
ADDRIS  072570
ADRERR  072351
AOE   = 001000
ASR1    001236
ATA   = 100000
ATABIT  101640
AT0   = 000001
AT1   = 000002
AT2   = 000004
AT3   = 000010
AT4   = 000020
AT5   = 000040
AT6   = 000100
AT7   = 000200
A16   = 000400
A17   = 001000
BADNO   072530
BADTMO  002160
BAI   = 000010
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BPTVEC= 000014
CHANGE  003236
CHGADR  001302
CKCLK   066060
CKCLK1  066130
CKCLK2  066172
CKCLK3  066202
CKERR   001250
CKSWR = 104407
CLOCK   066212
CLR   = 000040
CPSAVE  067220
CR    = 000015
CRLF  = 000200
DCK   = 100000
DDISP = 177570
DF1     101464
DF31    101506
DF36    101512
DF42    101515
DF5     101471
DF55    101517
DF56    101523
DF7     101477
DH1     077306
DH11    077750
DH13    100021
DH22    100141
DH23    100237
DH26    100337
DH3     077357
DH31    100416
DH36    100515
DH4     077426
DH42    100544
DH44    100563
DH46    100661
DH5     077551
DH55    100756
DH56    101034
DH57    101043
DH7     077625
DISPLA  001142
DISPRE  000174
DLT   = 100000
DMD   = 000001
DPE   = 000010
DPR   = 000400
DRQ   = 004000
DRY   = 000200
DSWR  = 177570
DTE   = 010000
DT00  = 000001
DT01  = 000002
DT02  = 000004
DT03  = 000010
DT04  = 000020
DT05  = 000040
DT06  = 000100
DT07  = 000200
DT08  = 000400
DT1     101176
DT13    101300
DT22    101316
DT23    101330
DT3     101212
DT31    101344
DT36    101356
DT37    101366
DT42    101376
DT46    101404
DT5     101226
DT54    101416
DT55    101430
DT56    101442
DT57    101446
DT6     101244
DT7     101260
DVA   = 004000
DVC   = 000200
ECH   = 000100
ECI   = 004000
EMTVEC= 000030
EM1     072654
EM10    073261
EM11    073311
EM12    073375
EM13    073445
EM14    073532
EM15    073575
EM16    073654
EM17    073727
EM2     072675
EM20    074006
EM21    074066
EM22    074141
EM23    074226
EM24    074274
EM25    074353
EM26    074416
EM27    074503
EM3     072717
EM30    074561
EM31    074656
EM32    074733
EM33    075004
EM34    075106
EM35    075211
EM36    075270
EM37    075342
EM4     073001
EM40    075410
EM41    075465
EM42    075527
EM43    075615
EM44    075672
EM45    075747
EM46    076021
EM47    076070
EM5     073032
EM50    076156
EM51    076240
EM52    076327
EM53    076422
EM54    076503
EM55    076576
EM56    076643
EM57    076706
EM6     073100
EM60    076747
EM61    077013
EM62    077045
EM63    077113
EM64    077161
EM65    077234
EM7     073200
ENTERA  072322
ERR   = 040000
ERROR = 104000
ERRVEC= 000004
EXEC    003062
FER   = 000020
FMT16 = 010000
F0    = 000002
F1    = 000004
F2    = 000010
F3    = 000020
F4    = 000040
GO    = 000001
GTSWR = 104406
HCE   = 000200
HCI   = 002000
HCRC  = 000400
HT    = 000011
IAE   = 002000
IBSAVE  067222
IE    = 000100
ILF   = 000001
ILR   = 000002
IOTVEC= 000020
IR    = 000100
IVC   = 010000
KYBCTL  001300
LBC   = 002000
LBT   = 002000
LF    = 000012
LSC   = 004000
MAXTN   101650
MCPE  = 020000
MDPE  = 000400
MOH   = 020000
MOL   = 010000
MUR   = 001000
MXF   = 001000
NBA   = 100000
NED   = 010000
NEM   = 004000
NOATA = 000001
NOCLOC  072452
NOSEIZ  001252
NTRH    072625
OFD   - 000200
OM    = 000001
OPE   = 020000
OPI   = 020000
OPPRT   001244
OR    = 000200
PAR   = 000010
PAT   = 000020
PFECH   067422
PFECH1  067432
PFECH2  067514
PFECH3  067546
PFECH4  067556
PFTSTN  067562
PGE   = 002000
PGM   = 001000
PIP   = 020000
PIRQ  = 177772
PIRQVE= 000240
PORTA   001224
PORTAI  072374
PORTB   001226
PORTBI  072423
PORTC   001230
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177776
PSEL  = 002000
PSW   = 177776
PTNBR   001240
PWRVEC= 000024
RDCHR = 104410
RDLIN = 104411
RDOCT = 104412
RDY   = 000200
RELERR  001254
RELOK = 000001
RESREG= 104414
RESVEC= 000010
RMAS  = 000016
RMBA  = 000004
RMCS1 = 000000
RMCS2 = 000010
RMDA  = 000006
RMDB  = 000022
RMDC  = 000034
RMDS  = 000012
RMDT  = 000026
RMEC1 = 000044
RMEC2 = 000046
RMER1 = 000014
RMER2 = 000042
RMLA  = 000020
```

SYMBOL TABLE

```
RMMR1  = 000024
RMMR2  = 000040
RMOF   = 000032
RMR    = 000004
RMSN   = 000030
RMWC   = 000002
RQA    = 100000
RQB    = 040000
RQSTA    001232
RQSTB    001234
R6     =%000006
R7     =%000007
SAVREG= 104413
SC     = 100000
SCOPE  = 000004
SC0    = 000100
SC1    = 000200
SC2    = 000400
SC3    = 001000
SC4    = 002000
SEIZPT   001242
SKI    = 100000
STACK  = 001100
START    002240
START1   002250
START2   002256
STKLMT= 177774
STOP     066650
SWR      001140
SWREG    000176
SW0    = 000001
SW00   = 000001
SW01   = 000002
SW02   = 000004
SW03   = 000010
SW04   = 000020
SW05   = 000040
SW06   = 000100
SW07   = 000200
SW08   = 000400
SW09   = 001000
SW1    = 000002
SW10   = 002000
SW11   = 004000
SW12   = 010000
SW13   = 020000
SW14   = 040000
SW15   = 100000
SW2    = 000004
SW3    = 000010
SW4    = 000020
SW5    = 000040
SW6    = 000100
SW7    = 000200
SW8    = 000400
SW9    = 001000
TAP    = 040000
TBITVE= 000014
TESTNO   072510
TEST1    003422
TEST10   013254
TEST11   014520
TEST12   015764
TEST13   017104
TEST14   020224
TEST15   021624
TEST16   023224
TEST17   024150
TEST2    004734
TEST20   025074
TEST21   026142
TEST22   027210
TEST23   031264
TEST24   032010
TEST25   033204
TEST26   034400
TEST27   036074
TEST3    006302
TEST30   037570
TEST31   041264
TEST32   042760
TEST33   044236
TEST34   045362
TEST35   046260
TEST36   047156
TEST37   050154
TEST4    007650
TEST40   051152
TEST41   053136
TEST42   055122
TEST43   056376
TEST44   057652
TEST45   060502
TEST46   061332
TEST47   062216
TEST5    011000
TEST50   063102
TEST51   064374
TEST6    012130
TEST7    012602
TIME     001256
TIMEA    001262
TIMEAM   001266
TIMEAP   001264
TIMEB    001270
TIMEBM   001274
TIMEBP   001272
TIMES    001276
TKVEC  = 000060
TOLER    066254
TPVEC  = 000064
TRAPVE= 000034
TRE    = 040000
TRTVEC= 000014
TSTADR   101524
TSTERR   072557
TSTNUM   001246
TST1     003364
TST1AA   003346
TST10    013216
TST11    014462
TST12    015726
TST13    017046
TST14    020166
TST15    021566
TST16    023166
TST17    024112
TST2     004676
TST20    025036
TST21    026104
TST22    027152
TST23    031226
TST24    031752
TST25    033146
TST26    034342
TST27    036036
TST3     006244
TST30    037532
TST31    041226
TST32    042722
TST33    044200
TST34    045324
TST35    046222
TST36    047120
TST37    050116
TST4     007612
TST40    051114
TST41    053100
TST42    055064
TST43    056340
TST44    057614
TST45    060444
TST46    061274
TST47    062160
TST5     010742
TST50    063044
TST50B   064114
TST51    064336
TST51B   065406
TST52    065630
TST6     012072
TST7     012544
TYPDS  = 104405
TYPE   = 104401
TYPOC  = 104402
TYPON  = 104404
TYPOS  = 104403
UNS    = 040000
UPE    = 020000
U0     = 000001
U1     = 000002
U3     = 000004
VV     = 000100
VVSET  = 000001
WATCH    001260
WCE    = 040000
WCF    = 000040
WLE    = 004000
WRL    = 004000
$AUTOB   001134
$BDADR   001122
$BDDAT   001126
$BELL    001202
$CHARC   070052
$CKSWR   071040
$CMTAG   001100
$CM1   = 000001
$CM2   = 000002
$CM3   = 000001
$CM4   = 000005
$CNTLC   071736
$CNTLG   071750
$CNTLU   071743
$CRLF    001207
$DBLK    070520
$DOAGN   066050
$DTBL    070510
$ENDAD   066040
$ENDCT   065676
$ENULL   066054
$EOP     065632
$EOPCT   065670
$ERFLG   001103
$ERMAX   001115
$ERROR   066676
$ERRPC   001116
$ERRTB   001310
$ERRTY   067224
$ERTTL   001112
$ESCAP   001200
$FILLC   001156
$FILLS   001155
$GDADR   001120
$GDDAT   001124
$GET42   066030
$GTSWR   071130
$GT42P   066024
$HD    = 000000
$HIOCT   072136
$ICNT    001104
$INTAG   001135
$ITEMB   001114
$LF      001210
$LKCSB   001214
$LKCSR   001212
$LKS     001220
$LLVEC   001222
$LPADR   001106
$LPERR   001110
$LPVEC   001216
$MNEW    071766
$MSWR    071755
$MXCNT   066646
$NULL    001154
$NWTST= 000000
$OCNT    070300
$OMODE   070302
$OVER    066632
$PASS    001100
$QUES    001206
$RDCHR   071402
$RDLIN   071472
$RDOCT   072000
$RDSZ  = 000010
$REGAD   001160
$REGO    001162
$RESRE   072176
$RMADR   001304
$RMVEC   001306
$RTNAD   066052
$SAVRE   072140
$SCOPE   066314
$SETUP= 000127
$STUP  = 177777
$SVLAD   066622
$SVPC  = 000210
$SWR   = 166000
$SWRMK= 000000
$TIMES   001176
$TKB     001146
$TKCNT   070530
$TKINT   070540
$TKQEN= 070537
$TKQIN   070532
$TKQOU   070534
$TKQSR   070536
$TKS     001144
$TKSRV   070610
$TMP0    001164
$TMP1    001166
$TMP2    001170
$TMP3    001172
$TMP4    001174
$TN    = 000053
$TPB     001152
$TPFLG   001157
$TPS     001150
$TRAP    072234
$TRAP2   072256
$TRP   = 000015
$TRPAD   072270
$TSTNM   001102
$TTYIN   071726
$TYPDS   070304
$TYPE    067564
```

SYMBOL TABLE

```
$TYPEC  067734          $TYPOC  070102          $TYPOS  070056          $XON  = 000021          $$GET4= 000000
$TYPEX  070054          $TYPON  070116          $XOFF = 000023          $XTSTR  066336          $OFILL  070301


. ABS.  101652     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  61952 WORDS  ( 242 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  71 PAGES
,A:CZRNHA/C=A:CZRNHA.DOC,CZRNHA,SYSMAC/M
```

CROSS REFERENCE TABLE (CREF V04.00)

$$GET4	10-1	10-1#												
$OFILL	16-1	16-1#	16-1*	16-1*										
$4OCAT	12-1	13-1												
$AUTOB	6-0#	9-28*	18-1	18-1	18-1									
$BDADR	6-0#	9-149*	9-149*	9-149*	9-149*	9-156*	9-156*	9-156*	9-156*	9-164*	9-164*	9-164*	9-164*	9-164*
	9-164*	9-164*	9-164*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*
	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*
	9-267*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*
	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-339*
	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*
	9-357*	9-384*	9-384*	9-397*	9-397*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*
	9-443*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-523*	9-523*	9-523*
	9-523*	9-523*	9-523*	9-523*	9-523*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-599*
	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-622*	9-622*	9-622*
	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-647*	9-647*	9-647*	9-647*	9-663*
	9-663*	9-663*	9-663*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-719*	9-719*	9-719*	9-719*	9-719*
	9-719*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-821*
	9-821*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-895*	9-895*	9-895*	9-895*	9-895*
	9-895*	9-895*	9-895*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-953*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-:30*
	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*
	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19*
	9-<19*	9-<19*	9-<19*	9-<36*	9-<36*	9-<36*	9-<36*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-=05*
	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->96*	9->96*	9->96*	9->96*	9->96*
	9->96*	9->96*	9->96*	9->96*	9->96*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?68*	9-?68*	9-?88*	9-?88*	9-@56*	9-@56*	9-@56*	9-@56*	9-@78*	9-@78*	9-@78*	9-@78*	9-A45*
	9-A45*	9-A45*	9-A45*	9-A46*	9-A46*	9-A46*	9-A46*	9-A46*	9-A46*	9-A46*	9-A46*	9-A75*	9-A75*	9-A75*
	9-A75*	9-A76*	9-A76*	9-A76*	9-A76*	9-A76*	9-A76*	9-A76*	9-A76*	24-32	24-33	24-34	24-35	24-36
	24-37	24-39												
$BDDAT	6-0#	9-149	9-149	9-149	9-149	9-149*	9-149*	9-156	9-156	9-156*	9-156*	9-164	9-164	9-164
	9-164	9-164	9-164	9-164	9-164	9-164*	9-164*	9-164*	9-164*	9-171*	9-172	9-267	9-267	9-267
	9-267	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*
	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-289	9-289	9-289	9-289	9-289*	9-289*	9-289*
	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*
	9-289*	9-289*	9-289*	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339*	9-339*	9-339*
	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357
	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-384	9-384	9-384*	9-384*	9-384*
	9-384*	9-384*	9-384*	9-397	9-397	9-397*	9-397*	9-397*	9-397*	9-397*	9-397*	9-443	9-443	9-443
	9-443	9-443	9-443	9-443	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*
	9-443*	9-443*	9-443*	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466*	9-466*	9-466*	9-466*
	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-523	9-523	9-523	9-523	9-523
	9-523	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-547	9-547	9-547	9-547
	9-547	9-547	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-599	9-599	9-599
	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599*	9-599*	9-599*	9-599*	9-599*
	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-622	9-622	9-622	9-622	9-622	9-622
	9-622	9-622	9-622	9-622	9-622	9-622	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*
	9-622*	9-622*	9-622*	9-622*	9-622*	9-647	9-647	9-647	9-647	9-647*	9-647*	9-647*	9-647*	9-647*
	9-647*	9-647*	9-647*	9-647*	9-663	9-663	9-663	9-663	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*
	9-663*	9-663*	9-663*	9-699	9-699	9-699	9-699	9-699	9-699	9-699*	9-699*	9-699*	9-699*	9-699*
	9-699*	9-699*	9-699*	9-699*	9-699*	9-719	9-719	9-719	9-719	9-719	9-719	9-719*	9-719*	9-719*
	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-821	9-821	9-821*	9-821*	9-821*	9-821*	9-821*

	9-821*	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*
	9-872*	9-872*	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895*	9-895*	9-895*	9-895*	9-895*
	9-895*	9-895*	9-895*	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953
	9-953	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975*	9-975*
	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*
	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05
	9-;05	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69	9-;69	9-;69	9-;69	9-;69
	9-;69	9-;69	9-;69	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19	9-<19	9-<19	9-<19*
	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<36	9-<36	9-<36	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*
	9-<36*	9-<36*	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*
	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99	9-=99
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24
	9->24	9->24	9->24	9->24	9->24	9->24	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96	9->96
	9->96	9->96	9->96	9->96	9->96	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*
	9->96*	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?68	9-?68	9-?68*	9-?68*	9-?68*	9-?68*	9-?88	9-?88
	9-?88*	9-?88*	9-?88*	9-?88*	9-@56	9-@56	9-@56	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*
	9-@78	9-@78	9-@78	9-@78*	9-@78*	9-@78*	9-@78*	9-@78*	9-@78*	9-@78*	9-A45	9-A45	9-A45	9-A45*
	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A46	9-A46	9-A46	9-A46	9-A46*	9-A46*	9-A46*	9-A46*	9-A75
	9-A75	9-A75	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A76	9-A76	9-A76	9-A76	9-A76*	9-A76*
	9-A76*	9-A76*	24-32	24-33	24-34	24-35	24-36	24-37	24-39					
$BELL	6-0#	13-1	13-1	13-1	18-1	18-1	18-1							
$CHARC	15-1	15-1#	15-1*	15-1*	15-1*									
$CKSWR	18-1#	21-1	21-1											
$CM1	6-0	6-0	6-0#	6-0#										
$CM2	6-0	6-0	6-0#	6-0/										
$CM3	6-0	6-0	6-0#											
$CM4	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0#	6-0#	6-0#	6-0#
	6-0#	6-0#												
$CMTAG	6-0#	9-23	9-23	9-23	9-23	9-23	9-23	9-23						
$CNTLC	18-1	18-1	18-1	18-1	18-1#									
$CNTLG	18-1	18-1#												
$CNTLU	18-1	18-1	18-1#											
$CRLF	6-0#	9-48	9-67	9-103	10-1	13-1	13-1	13-1	14-1	14-1	14-1	14-1	15-1	15-1
	15-1	18-1	18-1	18-1	18-1	19-1	19-1							
$DBLK	17-1	17-1	17-1#											
$DOAGN	10-1	10-1	10-1#											
$DTBL	17-1	17-1#												
$ENDAD	5-5	9-28	10-1#	13-1										
$ENDCT	9-23	10-1#												
$ENULL	10-1#													
$EOP	10-1#													
$EOPCT	9-23*	10-1	10-1#											
$ERFLG	6-0#	12-1	12-1	12-1	12-1	12-1*	13-1	13-1	13-1*					
$ERMAX	6-0#	9-23*	9-75	12-1	12-1									
$ERROR	9-23	13-1#												
$ERRPC	6-0#	13-1	13-1	13-1	13-1*	13-1*	14-1	14-1	24-32	24-33	24-34	24-35	24-36	24-37
	24-38	24-39	24-40	24-41	24-42	24-43	24-44	24-45	24-46	24-48				
$ERPTR	8-0#	14-1												

$ERRTY	13-1	14-1#												
$ERTTL	6-0#	10-1	10-1	10-1*	13-1	13-1	13-1*							
$ESCAP	6-0#	9-23*												
$FILLC	6-0#	15-1	15-1	15-1										
$FILLS	6-0#	15-1	15-1											
$GDADR	6-0#													
$GDDAT	6-0#	9-149	9-149	9-149*	9-149*	9-149*	9-149*	9-156	9-156	9-156*	9-156*	9-164	9-164	9-164
	9-164	9-164*	9-164*	9-164*	9-164*	9-164*	9-164*	9-164*	9-164*	9-169*	9-172	9-267	9-267	9-267
	9-267	9-267	9-267*	9-267*	9-267*	9-289	9-289	9-289	9-289	9-289	9-289*	9-289*	9-289*	9-339
	9-339	9-339	9-339	9-339	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-357	9-357	9-357
	9-357	9-357	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-384	9-384	9-384*	9-397	9-397
	9-397*	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443*	9-443*	9-443*	9-443*	9-443*	9-466
	9-466	9-466	9-466	9-466	9-466	9-466	9-466*	9-466*	9-466*	9-466*	9-466*	9-523	9-523	9-523
	9-523	9-523	9-523*	9-523*	9-523*	9-523*	9-523*	9-547	9-547	9-547	9-547	9-547	9-547*	9-547*
	9-547*	9-547*	9-547*	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599*	9-599*
	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-622	9-622	9-622	9-622	9-622
	9-622	9-622	9-622	9-622	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*
	9-622*	9-647	9-647	9-647	9-647	9-647	9-647*	9-647*	9-647*	9-663	9-663	9-663	9-663	9-663
	9-663*	9-663*	9-663*	9-699	9-699	9-699	9-699	9-699	9-699	9-699*	9-699*	9-699*	9-699*	9-699*
	9-719	9-719	9-719	9-719	9-719	9-719	9-719*	9-719*	9-719*	9-719*	9-719*	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-821
	9-821	9-821*	9-872	9-872	9-872	9-872	9-872	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-895
	9-895	9-895	9-895	9-895	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-953	9-953	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-953*	9-953*	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975*	9-975*	9-975*
	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*
	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05*	9-;05*
	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69	9-;69	9-;69	9-;69	9-;69*	9-;69*	9-;69*
	9-;69*	9-;69*	9-;69*	9-;69*	9-<19	9-<19	9-<19	9-<19*	9-<19*	9-<36	9-<36	9-<36	9-<36*	9-<36*
	9-<85	9-<85	9-<85	9-<85	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-=05	9-=05	9-=05	9-=05	9-=05*
	9-=05*	9-=05*	9-=05*	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96	9->96
	9->96	9->96	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9-?14	9-?14	9-?14	9-?14
	9-?14	9-?14	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?68	9-?68	9-?68*	9-?88
	9-?88	9-?88*	9-@56	9-@56	9-@56	9-@56*	9-@56*	9-@78	9-@78	9-@78	9-@78*	9-@78*	9-A45	9-A45
	9-A45	9-A45*	9-A45*	9-A46	9-A46	9-A46*	9-A46*	9-A75	9-A75	9-A75	9-A75*	9-A75*	9-A76	9-A76
	9-A76*	9-A76*	24-33	24-34	24-36									
$GET42	10-1#													
$GT42P	10-1	10-1#												
$GTSWR	18-1#	21-1	21-1											
$HD	4-592	4-592	4-592											
$HIOCT	19-1#	19-1*												
$ICNT	6-0#	9-91*	12-1	12-1	12-1	12-1*	12-1*							
$INTAG	6-0#	18-1	18-1	18-1	18-1	18-1*								
$ITEMB	6-0#	13-1	13-1	13-1	13-1	13-1*	13-1*	14-1						
$LF	6-0#	13-1	13-1	15-1	15-1	18-1	18-1	18-1	19-1	19-1				
$LKCSB	7-0#	11-12*												
$LKCSR	7-0#	11-8	11-13*											
$LKS	7-0#	11-17	11-21*											
$LLVEC	7-0#	11-18												
$LPADR	6-0#	9-23*	9-76*	9-136*	9-198*	9-287*	9-305*	9-355*	9-372*	9-395*	9-418*	9-464*	9-488*	9-545*
	9-568*	9-620*	9-636*	9-661*	9-681*	9-717*	9-736*	9-805*	9-842*	9-893*	9-919*	9-973*	9-996*	9-:51*

	9-:69*	9-;24*	9-;84*	9-<34*	9-<55*	9-=03*	9-=57*	9->22*	9->44*	9-?12*	9-?32*	9-?86*	9-@11*	9-@76*
	9-A05*	9-A73*	12-1	12-1	12-1	12-1*								
$LPERR	6-0#	9-23*	9-77*	9-136*	9-198*	9-287*	9-305*	9-355*	9-372*	9-395*	9-418*	9-464*	9-488*	9-545*
	9-568*	9-620*	9-636*	9-661*	9-681*	9-717*	9-736*	9-805*	9-842*	9-893*	9-919*	9-973*	9-996*	9-:51*
	9-:69*	9-;24*	9-;84*	9-<19*	9-<34*	9-<36*	9-<55*	9-=03*	9-=57*	9->22*	9->44*	9-?12*	9-?32*	9-?86*
	9-@11*	9-@76*	9-A05*	9-A73*										
$LPVEC	7-0#	11-9												
$MAIL	9-23	9-28	12-1	13-1	15-1									
$MNEW	18-1	18-1#												
$MSWR	18-1	18-1#												
$MXCNT	12-1	12-1	12-1	12-1#										
$NULL	6-0#	15-1	15-1	15-1										
$NWTST	9-136	9-136	9-136#	9-136#	9-198	9-198	9-198#	9-198#	9-287	9-287	9-287#	9-287#	9-305	9-305
	9-305#	9-305#	9-355	9-355	9-355#	9-355#	9-372	9-372	9-372#	9-372#	9-395	9-395	9-395#	9-395#
	9-418	9-418	9-418#	9-418#	9-464	9-464	9-464#	9-464#	9-488	9-488	9-488#	9-488#	9-545	9-545
	9-545#	9-545#	9-568	9-568	9-568#	9-568#	9-620	9-620	9-620#	9-620#	9-636	9-636	9-636#	9-636#
	9-661	9-661	9-661#	9-661#	9-681	9-681	9-681#	9-681#	9-717	9-717	9-717#	9-717#	9-736	9-736
	9-736#	9-736#	9-805	9-805	9-805#	9-805#	9-842	9-842	9-842#	9-842#	9-893	9-893	9-893#	9-893#
	9-919	9-919	9-919#	9-919#	9-973	9-973	9-973#	9-973#	9-996	9-996	9-996#	9-996#	9-:51	9-:51
	9-:51#	9-:51#	9-:69	9-:69	9-:69#	9-:69#	9-;24	9-;24	9-;24#	9-;24#	9-;84	9-;84	9-;84#	9-;84#
	9-<34	9-<34	9-<34#	9-<34#	9-<55	9-<55	9-<55#	9-<55#	9-=03	9-=03	9-=03#	9-=03#	9-=57	9-=57
	9-=57#	9-=57#	9->22	9->22	9->22#	9->22#	9->44	9->44	9->44#	9->44#	9-?12	9-?12	9-?12#	9-?12#
	9-?32	9-?32	9-?32#	9-?32#	9-?86	9-?86	9-?8[illegible]	9-?86#	9-@11	9-@11	9-@11#	9-@11#	9-@76	9-@76
	9-@76#	9-@76#	9-A05	9-A05	9-A05#	9-A05#	9-A73	9-A73	9-A73#	9-A73#	9-A80#			
$OCNT	16-1#	16-1*	16-1*											
$OMODE	16-1	16-1#	16-1*	16-1*	16-1*	16-1*								
$OVER	12-1	12-1	12-1	12-1#										
$PASS	6-0#	9-74*	10-1	10-1	10-1	10-1*	10-1*	12-1	12-1	12-1				
$QUES	6-0#	13-1	13-1	15-1	15-1	18-1	18-1	18-1	18-1	19-1	19-1	19-1		
$R2A	21-1													
$RDCHR	18-1#	21-1	21-1											
$RDDEC	21-1													
$RDLIN	18-1#	21-1	21-1											
$RDOCT	19-1#	21-1	21-1											
$RDSZ	18-1	18-1#												
$REGO	6-0#													
$REGAD	6-0#													
$RESRE	20-1#	21-1												
$RMADR	7-0#	9-68	9-101	9-108*	9-111	9-133	24-47							
$RMVEC	7-0#													
$RTNAD	10-1#													
$SAVRE	20-1#	21-1	21-1											
$SCOPE	9-23	12-1#												
$SETUP	4-777	4-777	4-777	4-777	4-777	4-777#	4-777#	4-777#	4-777#	4-777#	4-777#	9-23	9-23	9-23
	9-23	9-23	9-23	9-23	9-23	9-23	9-23	9-23	9-23	9-28	9-28	9-28	10-1	10-1
	12-1	13-1	13-1	13-1	13-1	18-1	18-1	18-1	18-1	18-1				
$STUP	4-777	4-777	4-777	4-777	4-777	4-777#	4-777#	4-777#	4-777#	4-777#	4-777#	4-777#	4-777#	4-777#
	4-777#													
$SVLAD	12-1	12-1#												
$SVPC	5-5	5-5#												
$SWR	4-588#	4-592	4-593	4-593	4-593	4-593	4-593	4-593	4-593	4-593	6-0	6-0	6-0	9-23
	9-23	9-23	9-23	9-23	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488
	9-545	9-568	9-620	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996
	9-:51	9-:69	9-;24	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11
	9-@76	9-A05	9-A73	9-A80	10-1	10-1	10-1	10-1	10-1	11-53#	12-1	12-1	12-1	12-1
	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	12-1	13-1

	13-1	13-1	13-1	13-1	13-1	13-1	13-1	13-1	13-1	13-1				
$SWRMK	12-1													
$TESTN	14-1													
$TIMES	6-0#	9-23*	9-136*	9-198*	9-287*	9-305*	9-355*	9-372*	9-395*	9-418*	9-464*	9-488*	9-545*	9-568*
	9-620*	9-636*	9-661*	9-681*	9-717*	9-736*	9-805*	9-842*	9-893*	9-919*	9-973*	9-996*	9-:51*	9-:69*
	9-;24*	9-;84*	9-<34*	9-<55*	9-=03*	9-=57*	9->22*	9->44*	9-?12*	9-?32*	9-?86*	9-@11*	9-@76*	9-A05*
	9-A73*	10-1*	12-1	12-1	12-1	12-1*	12-1*							
$TKB	6-0#	15-1	15-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1				
$TKCNT	18-1	18-1	18-1#	18-1*	18-1*	18-1*								
$TKINT	9-29	9-72	18-1	18-1	18-1#									
$TKQEN	18-1	18-1	18-1#											
$TKQIN	18-1	18-1	18-1#	18-1*	18-1*	18-1*	18-1*							
$TKQOU	18-1	18-1	18-1#	18-1*	18-1*	18-1*								
$TKQSR	18-1	18-1	18-1	18-1#										
$TKS	6-0#	15-1	15-1	18-1	18-1	18-1	18-1	18-1	18-1*	18-1*	18-1*	18-1*	18-1*	18-1*
$TKSRV	18-1	18-1#												
$TMP0	6-0#	9-149	9-149	9-149*	9-149*	9-149*	9-149*	9-164	9-164	9-164	9-164	9-164*	9-164*	9-164*
	9-164*	9-164*	9-164*	9-164*	9-164*	9-267	9-267	9-267	9-267	9-267*	9-267*	9-267*	9-267*	9-289
	9-289	9-289	9-289	9-289*	9-289*	9-289*	9-289*	9-339	9-339	9-339	9-339	9-339	9-339	9-339*
	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-357	9-357	9-357	9-357	9-357	9-357	9-357*
	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-384	9-384	9-384	9-384*	9-384*	9-397	9-397
	9-397	9-397*	9-397*	9-443	9-443	9-443	9-443	9-443	9-443	9-443*	9-443*	9-443*	9-443*	9-443*
	9-466	9-466	9-466	9-466	9-466	9-466	9-466*	9-466*	9-466*	9-466*	9-466*	9-523	9-523	9-523
	9-523	9-523*	9-523*	9-523*	9-523*	9-547	9-547	9-547	9-547	9-547*	9-547*	9-547*	9-547*	9-599
	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*
	9-599*	9-599*	9-599*	9-599*	9-599*	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622*
	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-647	9-647	9-647
	9-647	9-647*	9-647*	9-647*	9-647*	9-663	9-663	9-663	9-663	9-663*	9-663*	9-663*	9-663*	9-699
	9-699	9-699	9-699	9-699	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-719	9-719	9-719	9-719
	9-719	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-821	9-821	9-821	9-821	9-821*	9-821*	9-821*	9-821*	9-872	9-872	9-872	9-872	9-872
	9-872	9-872	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-895	9-895	9-895	9-895	9-895
	9-895	9-895	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-953	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-953*	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975*	9-975*	9-975*	9-975*	9-975*
	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30
	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05*	9-;05*
	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69*
	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19	9-<19	9-<19	9-<19	9-<19*	9-<19*	9-<19*	9-<19*	9-<36
	9-<36	9-<36	9-<36	9-<36*	9-<36*	9-<36*	9-<36*	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85*	9-<85*
	9-<85*	9-<85*	9-<85*	9-<85*	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*
	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24*
	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->96	9->96	9->96	9->96	9->96	9->96	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*
	9->96*	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?68	9-?68	9-?68	9-?68*	9-?68*	9-?88	9-?88	9-?88	9-?88*	9-?88*	9-@56	9-@56	9-@56
	9-@56*	9-@56*	9-@78	9-@78	9-@78	9-@78*	9-@78*	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45*	9-A45*
	9-A45*	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75*	9-A75*	9-A75*	24-48				
$TMP1	6-0#	9-267	9-267	9-267	9-267*	9-267*	9-267*	9-267*	9-289	9-289	9-289	9-289*	9-289*	9-289*
	9-289*	9-339	9-339	9-339*	9-339*	9-357	9-357	9-357*	9-357*	9-384	9-384	9-384*	9-384*	9-397
	9-397	9-397*	9-397*	9-443	9-443	9-443	9-443*	9-443*	9-443*	9-443*	9-466	9-466	9-466	9-466*

	9-466*	9-466*	9-466*	9-523	9-523	9-523*	9-523*	9-547	9-547	9-547*	9-547*	9-599	9-599	9-599
	9-599*	9-599*	9-599*	9-599*	9-622	9-622	9-622	9-622*	9-622*	9-622*	9-622*	9-647	9-647	9-647
	9-647*	9-647*	9-647*	9-647*	9-663	9-663	9-663	9-663*	9-663*	9-663*	9-663*	9-699	9-699	9-699
	9-699*	9-699*	9-699*	9-699*	9-719	9-719	9-719	9-719*	9-719*	9-719*	9-719*	9-791	9-791	9-791
	9-791	9-791*	9-791*	9-791*	9-791*	9-821	9-821	9-821*	9-821*	9-872	9-872	9-872*	9-872*	9-895
	9-895	9-895*	9-895*	9-953	9-953	9-953	9-953*	9-953*	9-953*	9-953*	9-975	9-975	9-975	9-975*
	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:53	9-:53	9-:53	9-:53*
	9-:53*	9-:53*	9-:53*	9-;05	9-;05	9-;05*	9-;05*	9-;69	9-;69	9-;69*	9-;69*	9-<19	9-<19	9-<19
	9-<19*	9-<19*	9-<19*	9-<19*	9-<36	9-<36	9-<36	9-<36*	9-<36*	9-<36*	9-<36*	9-<85	9-<85	9-<85*
	9-<85*	9-=05	9-=05	9-=05*	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99*	9-=99*	9-=99*	9-=99*	9->24
	9->24	9->24	9->24	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96*	9->96*	9-?14	9-?14	9-?14*
	9-?14*	9-?68	9-?68	9-?68*	9-?68*	9-?88	9-?88	9-?88*	9-?88*	9-@56	9-@56	9-@56*	9-@56*	9-@78
	9-@78	9-@78*	9-@78*	9-A45	9-A45	9-A45*	9-A45*	9-A75	9-A75	9-A75*	9-A75*	24-48		
$TMP2	6-0#	9-267	9-267	9-267	9-267*	9-267*	9-289	9-289	9-289	9-289*	9-289*	9-339	9-339	9-339
	9-339*	9-339*	9-357	9-357	9-357	9-357*	9-357*	9-384	9-384	9-384	9-384*	9-384*	9-397	9-397
	9-397	9-397*	9-397*	9-443	9-443	9-443	9-443*	9-443*	9-466	9-466	9-466	9-466*	9-466*	9-523
	9-523	9-523	9-523*	9-523*	9-547	9-547	9-547	9-547*	9-547*	9-599	9-599	9-599	9-599*	9-599*
	9-622	9-622	9-622	9-622*	9-622*	9-647	9-647	9-647	9-647*	9-647*	9-663	9-663	9-663	9-663*
	9-663*	9-699	9-699	9-699	9-699*	9-699*	9-719	9-719	9-719	9-719*	9-719*	9-791	9-791	9-791
	9-791	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-821	9-821	9-821	9-821*	9-821*	9-872	9-872
	9-872	9-872*	9-872*	9-895	9-895	9-895	9-895*	9-895*	9-953	9-953	9-953	9-953*	9-953*	9-975
	9-975	9-975	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:53	9-:53	9-:53	9-:53*	9-:53*
	9-;05	9-;05	9-;05	9-;05*	9-;05*	9-;69	9-;69	9-;69	9-;69*	9-;69*	9-<19	9-<19	9-<19	9-<19*
	9-<19*	9-<36	9-<36	9-<36	9-<36*	9-<36*	9-<85	9-<85	9-<85	9-<85*	9-<85*	9-=05	9-=05	9-=05
	9-=05*	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99*	9-=99*	9-=99*	9-=99*	9->24	9->24
	9->24	9->24	9->24	9->24	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96	9->96*	9->96*	9-?14
	9-?14	9-?14	9-?14*	9-?14*	9-?68	9-?68	9-?68	9-?68*	9-?68*	9-?88	9-?88	9-?88	9-?88*	9-?88*
	9-@56	9-@56	9-@56	9-@56*	9-@56*	9-@78	9-@78	9-@78	9-@78*	9-@78*	9-A45	9-A45	9-A45	9-A45*
	9-A45*	9-A75	9-A75	9-A75	9-A75*	9-A75*	24-44	24-48						
$TMP3	6-0#	9-267	9-267	9-267	9-267*	9-267*	9-289	9-289	9-289	9-289*	9-289*	9-339	9-339	9-339
	9-339*	9-339*	9-357	9-357	9-357	9-357*	9-357*	9-384	9-384	9-384	9-384*	9-384*	9-397	9-397
	9-397	9-397*	9-397*	9-443	9-443	9-443	9-443*	9-443*	9-466	9-466	9-466	9-466*	9-466*	9-523
	9-523	9-523	9-523*	9-523*	9-547	9-547	9-547	9-547*	9-547*	9-599	9-599	9-599	9-599*	9-599*
	9-622	9-622	9-622	9-622*	9-622*	9-647	9-647	9-647	9-647*	9-647*	9-663	9-663	9-663	9-663*
	9-663*	9-699	9-699	9-699	9-699*	9-699*	9-719	9-719	9-719	9-719*	9-719*	9-791	9-791	9-791
	9-791	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-821	9-821	9-821	9-821*	9-821*	9-872	9-872
	9-872	9-872*	9-872*	9-895	9-895	9-895	9-895*	9-895*	9-953	9-953	9-953	9-953*	9-953*	9-975
	9-975	9-975	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:53	9-:53	9-:53	9-:53*	9-:53*
	9-;05	9-;05	9-;05	9-;05*	9-;05*	9-;69	9-;69	9-;69	9-;69*	9-;69*	9-<19	9-<19	9-<19	9-<19*
	9-<19*	9-<36	9-<36	9-<36	9-<36*	9-<36*	9-<85	9-<85	9-<85	9-<85*	9-<85*	9-=05	9-=05	9-=05
	9-=05*	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99*	9-=99*	9-=99*	9-=99*	9->24	9->24
	9->24	9->24	9->24	9->24	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96	9->96*	9->96*	9-?14
	9-?14	9-?14	9-?14*	9-?14*	9-?68	9-?68	9-?68	9-?68*	9-?68*	9-?88	9-?88	9-?88	9-?88*	9-?88*
	9-@56	9-@56	9-@56	9-@56*	9-@56*	9-@78	9-@78	9-@78	9-@78*	9-@78*	9-A45	9-A45	9-A45	9-A45*
	9-A45*	9-A75	9-A75	9-A75	9-A75*	9-A75*	24-44	24-48						
$TMP4	6-0#	9-149	9-149	9-149*	9-149*	9-149*	9-149*	9-164	9-164	9-164	9-164	9-164*	9-164*	9-164*
	9-164*	9-164*	9-164*	9-164*	9-164*	9-339	9-339	9-339	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*
	9-357	9-357	9-357	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-523	9-523*	9-523*	9-547	9-547*
	9-547*	9-599	9-599	9-599	9-599	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-622
	9-622	9-622	9-622	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-699	9-699*	9-699*
	9-719	9-719*	9-719*	9-791	9-791	9-791*	9-791*	9-791*	9-791*	9-872	9-872	9-872*	9-872*	9-872*
	9-872*	9-895	9-895	9-895*	9-895*	9-895*	9-895*	9-953	9-953	9-953	9-953	9-953*	9-953*	9-953*
	9-953*	9-953*	9-953*	9-953*	9-953*	9-975	9-975	9-975	9-975	9-975*	9-975*	9-975*	9-975*	9-975*
	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*
	9-:30*	9-:53	9-:53	9-:53	9-:53	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05
	9-;05	9-;05	9-;05	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69	9-;69	9-;69*

	9-:69*	9-:69*	9-:69*	9-<85	9-<85	9-<85*	9-<85*	9-<85*	9-<85*	9-=05	9-=05	9-=05*	9-=05*	9-=05*
	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9->24	9->24	9->24	9->24	9->24	9->24	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->96	9->96	9->96	9->96*	9->96*
	9->96*	9->96*	9->96*	9->96*	9-?14	9-?14	9-?14	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	
$TN	4-589#	4-592	9-136	9-136	9-136	9-136#	9-198	9-198	9-198	9-198#	9-287	9-287	9-287	9-287#
	9-305	9-305	9-305	9-305#	9-355	9-355	9-355	9-355#	9-372	9-372	9-372	9-372#	9-395	9-395
	9-395	9-395#	9-418	9-418	9-418	9-418#	9-464	9-464	9-464	9-464#	9-488	9-488	9-488	9-488#
	9-545	9-545	9-545	9-545#	9-568	9-568	9-568	9-568#	9-620	9-620	9-620	9-620#	9-636	9-636
	9-636	9-636#	9-661	9-661	9-661	9-661#	9-681	9-681	9-681	9-681#	9-717	9-717	9-717	9-717#
	9-736	9-736	9-736	9-736#	9-805	9-805	9-805	9-805#	9-842	9-842	9-842	9-842#	9-893	9-893
	9-893	9-893#	9-919	9-919	9-919	9-919#	9-973	9-973	9-973	9-973#	9-996	9-996	9-996	9-996#
	9-:51	9-:51	9-:51	9-:51#	9-:69	9-:69	9-:69	9-:69#	9-;24	9-;24	9-;24	9-;24#	9-;84	9-;84
	9-;84	9-;84#	9-<34	9-<34	9-<34	9-<34#	9-<55	9-<55	9-<55	9-<55#	9-=03	9-=03	9-=03	9-=03#
	9-=57	9-=57	9-=57	9-=57#	9->22	9->22	9->22	9->22#	9->44	9->44	9->44	9->44#	9-?12	9-?12
	9-?12	9-?12#	9-?32	9-?32	9-?32	9-?32#	9-?86	9-?86	9-?86	9-?86#	9-@11	9-@11	9-@11	9-@11#
	9-@76	9-@76	9-@76	9-@76#	9-A05	9-A05	9-A05	9-A05#	9-A45	9-A46	9-A46	9-A73	9-A73	9-A73
	9-A73#	9-A75	9-A76	9-A76	9-A80	9-A80	9-A80#	25-27						
$TPB	6-0#	15-1	15-1	15-1*										
$TPFLG	6-0#	15-1	15-1	15-1										
$TPS	6-0#	15-1	15-1	15-1										
$TRAP	9-23	21-1#												
$TRAP2	21-1	21-1#												
$TRP	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1
	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1
	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1
	21-1	21-1	21-1	21-1	21-1	21-1	21-1#	21-1#	21-1#	21-1#	21-1#	21-1#	21-1#	21-1#
	21-1#	21-1#	21-1#	21-1#	21-1#									
$TRPAD	21-1	21-1#												
$TSTNM	6-0#	9-136*	9-198*	9-287*	9-305*	9-355*	9-372*	9-395*	9-418*	9-464*	9-488*	9-545*	9-568*	9-620*
	9-636*	9-661*	9-681*	9-717*	9-736*	9-805*	9-842*	9-893*	9-919*	9-973*	9-996*	9-:51*	9-:69*	9-;24*
	9-;84*	9-<34*	9-<55*	9-=03*	9-=57*	9->22*	9->44*	9-?12*	9-?32*	9-?86*	9-@11*	9-@76*	9-A05*	9-A73*
	10-1*	12-1	12-1	12-1	12-1	12-1*	13-1	13-1	13-1	13-1	14-1			
$TTYIN	18-1	18-1	18-1	18-1	18-1	18-1#								
$TYPBN	21-1													
$TYPDS	17-1#	21-1	21-1											
$TYPE	15-1#	21-1	21-1											
$TYPEC	15-1	15-1	15-1	15-1#	18-1									
$TYPEX	15-1	15-1	15-1#											
$TYPOC	16-1#	21-1	21-1											
$TYPON	16-1	16-1#	21-1											
$TYPOS	16-1#	21-1												
$XOFF	15-1	15-1												
$XON	15-1	15-1	18-1											
$XTSTR	12-1#													
A16	4-603#													
A17	4-604#													
ADDRIS	9-100	22-11#												
ADRERR	9-36	22-4#												
AOE	4-673#													
ASR1	7-0#	9-53*	9-791	9-821	9-;69	9-;69	9-;69	9-<19	9-<19	9-<36	9-<36	9-A45	9-A75	
AT0	4-690#													
AT1	4-691#													
AT2	4-692#													
AT3	4-693#													
AT4	4-694#													

AT5	4-695#													
AT6	4-696#													
AT7	4-697#													
ATA	4-660#	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-339	9-339	9-339	9-339	9-339
	9-339	9-339	9-339	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-384	9-384	9-384
	9-384	9-397	9-397	9-397	9-397	9-443	9-443	9-443	9-443	9-443	9-466	9-466	9-466	9-466
	9-466	9-523	9-523	9-523	9-523	9-547	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-622
	9-622	9-622	9-622	9-647	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-699	9-699	9-699
	9-699	9-719	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-821	9-821	9-821	9-821	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-895
	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-953	9-953	9-953	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-975	9-975	9-975	9-975	9-975	9-975
	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05	9-;05
	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69	9-;69	9-;69	9-;69	9-;69
	9-;69	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36	9-<36	9-<85	9-<85	9-<85
	9-<85	9-<85	9-<85	9-<85	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=99	9-=99	9-=99
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->96	9->96	9->96
	9->96	9->96	9->96	9->96	9->96	9->96	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14
	9-?14	9-?68	9-?68	9-?88	9-?88	9-a56	9-a56	9-a56	9-a56	9-a78	9-a78	9-a78	9-a78	9-A45
	9-A45	9-A45	9-A75	9-A75	9-A75									
ATABIT	9-53	25-18#												
BADNO	9-86	22-9#												
BADTMO	9-3#	9-25												
BAI	4-615#													
BIT0	4-595#													
BIT00	4-595	4-595#	4-650	12-1	12-1	13-1	13-1							
BIT01	4-595	4-595#												
BIT02	4-595	4-595#												
BIT03	4-595	4-595#												
BIT04	4-595	4-595#												
BIT05	4-595	4-595#												
BIT06	4-595	4-595#												
BIT07	4-595	4-595#												
BIT08	4-595	4-595#												
BIT09	4-595	4-595#	4-684	12-1	13-1									
BIT1	4-595#													
BIT10	4-595#	13-1												
BIT11	4-595#	12-1												
BIT12	4-595#													
BIT13	4-595#	13-1												
BIT14	4-595#	4-685	12-1											
BIT15	4-595#	4-686												
BIT2	4-595#													
BIT3	4-595#													
BIT4	4-595#													
BIT5	4-595#	4-646												
BIT6	4-595#													
BIT7	4-595#													
BIT8	4-595#													
BIT9	4-595#													
BPTVEC	4-595#													
CHANGE	9-30	9-97#												
CHGADR	7-0#	9-15*	9-17*	9-97	9-99*									

CKCLK	9-57	9-70	11-6#											
CKCLK1	11-6	11-15#												
CKCLK2	11-16	11-23#												
CKCLK3	11-14	11-22	11-25#											
CKERR	7-0#	9-149	9-149	9-149*	9-149*	9-149*	9-149*	9-156*	9-156*	9-156*	9-156*	9-164*	9-164*	9-164*
	9-164*	9-164*	9-164*	9-164*	9-164*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-357*	9-357*	9-357*
	9-357*	9-357*	9-357*	9-443	9-443*	9-443*	9-466	9-466*	9-466*	9-523	9-523*	9-523*	9-523*	9-523*
	9-523*	9-523*	9-547	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-599*	9-599*	9-599*	9-599*	9-599*
	9-599*	9-599*	9-599*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-699*	9-699*	9-719*
	9-719*	9-791*	9-791*	9-791*	9-791*	9-872*	9-872*	9-872*	9-872*	9-895*	9-895*	9-895*	9-895*	9-953
	9-953	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-975	9-975	9-975*	9-975*	9-975*
	9-975*	9-975*	9-975*	9-975*	9-975*	9-:30	9-:30	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*
	9-:30*	9-:53	9-:53	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05	9-;05	9-;05*
	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;69	9-;69	9-;69*	9-;69*	9-;69*	9-;69*	9-<85*
	9-<85*	9-<85*	9-<85*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-@56	9-@56*	9-@56*	9-@78	9-@78*	9-@78*	9-A46*	9-A46*	9-A46*
	9-A46*	9-A76*	9-A76*	9-A76*	9-A76*									
CKSWR	12-1	13-1	13-1	21-1#										
CLOCK	11-10	11-19	11-30#											
CLR	4-617#	4-646#	9-138	9-149	9-149	9-339	9-357	9-523	9-547	9-599	9-622	9-;05		
CPSAVE	12-1	12-1	12-1	12-1*	12-1*	13-1	13-1	13-1	13-1	13-1	13-1#	13-1*	13-1*	14-1
CR	4-595#	15-1	15-1	24-3	24-6	24-9	24-11	24-13	24-15	24-17	24-21	24-23		
CRLF	4-595#	9-6	9-28	9-28	15-1	15-1	22-3	22-4	22-5	22-6	22-7	22-7	22-9	22-10
	22-11													
DCK	4-679#													
DDISP	4-595#	6-0	9-23											
DF1	8-7	8-14	8-21	8-63	8-133	8-140	8-226	8-233	8-261	8-275	8-303	8-346	24-50#	
DF31	8-126	8-156	8-177	8-254	8-268	8-310	24-53#							
DF36	8-70	8-112	8-119	8-148	8-170	8-191	8-198	8-205	8-212	8-219	8-360	24-54#		
DF42	8-240	8-247	24-55#											
DF5	8-35	8-42	8-56	8-77	8-84	8-184	8-282	8-289	8-296	8-332	8-339	8-353	8-374	24-51#
DF55	8-317	24-56#												
DF56	8-324	24-57#												
DF7	8-28	8-49	8-91	8-98	8-105	8-163	8-367	24-52#						
DH1	8-5	8-12	8-224	8-259	24-1#									
DH11	8-61	24-8#												
DH13	8-40	8-75	8-82	8-280	8-294	8-351	24-9#							
DH22	8-124	8-154	8-308	24-11#										
DH23	8-131	8-138	8-231	8-273	8-301	8-344	24-13#							
DH26	24-15#													
DH3	8-19	24-2#												
DH31	8-175	24-17#												
DH36	8-68	8-110	8-117	8-146	8-168	8-189	8-196	8-203	8-210	8-217	8-358	24-19#		
DH4	8-26	24-3#												
DH42	8-238	8-245	24-20#											
DH44	8-252	24-21#												
DH46	8-266	24-23#												
DH5	8-33	8-54	8-182	8-287	8-337	8-372	24-5#							
DH55	8-315	24-25#												
DH56	8-322	24-26#												
DH57	8-330	24-27#												
DH7	8-47	8-89	8-96	8-103	8-161	8-365	24-6#							
DISPLA	6-0#	9-23*	9-23*	12-1*	13-1*									
DISPRE	5-1#	9-23												

DLT	4-627#													
DMD	4-683#	9-=99	9-=99	9->24	9->24									
DPE	4-728#													
DPR	4-653#	9-164	9-164	9-267	9-267	9-289	9-289	9-339	9-357	9-384	9-397	9-443	9-443	9-443
	9-466	9-466	9-466	9-523	9-547	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-647
	9-647	9-663	9-663	9-699	9-699	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-821
	9-872	9-872	9-895	9-895	9-953	9-953	9-975	9-975	9-:30	9-:30	9-:53	9-:53	9-<19	9-<36
	9-<85	9-=05	9-=99	9-=99	9-=99	9->24	9->24	9->24	9-?68	9-?88	9-@56	9-@78	9-A45	9-A45
	9-A75	9-A75												
DRQ	4-713#													
DRY	4-652#	9-164	9-164	9-267	9-267	9-289	9-289	9-339	9-357	9-384	9-397	9-443	9-443	9-443
	9-466	9-466	9-466	9-523	9-547	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-647
	9-647	9-663	9-663	9-699	9-699	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-821
	9-872	9-872	9-895	9-895	9-953	9-953	9-975	9-975	9-:30	9-:30	9-:53	9-:53	9-<19	9-<36
	9-<85	9-<85	9-=05	9-=05	9-=99	9-=99	9-=99	9->24	9->24	9->24	9-?68	9-?88	9-@56	9-@78
	9-A45	9-A45	9-A75	9-A75										
DSWR	4-595#	6-0	9-23											
DT00	4-704#													
DT01	4-705#													
DT02	4-706#													
DT03	4-707#													
DT04	4-708#													
DT05	4-709#													
DT06	4-710#													
DT07	4-711#													
DT08	4-712#													
DT1	8-6	8-13	8-62	8-260	8-302	24-32#								
DT13	8-41	8-76	8-83	8-281	8-295	8-352	24-37#							
DT22	8-125	8-155	24-38#											
DT23	8-132	8-139	8-225	8-232	8-274	8-345	24-39#							
DT3	8-20	24-33#												
DT31	8-176	8-253	24-40#											
DT36	8-147	8-169	8-190	8-197	8-211	8-359	24-41#							
DT37	8-69	8-111	8-118	8-204	8-218	24-42#								
DT42	8-239	8-246	24-43#											
DT46	8-267	24-44#												
DT5	8-34	8-55	8-183	8-288	8-338	8-373	24-34#							
DT54	8-309	24-45#												
DT55	8-316	24-46#												
DT56	8-323	24-47#												
DT57	8-331	24-48#												
DT6	24-35#													
DT7	8-27	8-48	8-90	8-97	8-104	8-162	8-366	24-36#						
DTE	4-676#													
DVA	4-642#													
DVC	4-729#													
ECH	4-670#													
ECI	4-740#													
EM1	8-4	23-3#												
EM10	8-53	23-10#												
EM11	8-60	23-11#												
EM12	8-67	23-12#												
EM13	8-74	23-13#												
EM14	8-81	23-14#												
EM15	8-88	23-15#												
EM16	8-95	23-16#												

EM17	8-102	23-17#												
EM2	8-11	23-4#												
EM20	8-109	23-18#												
EM21	8-116	23-19#												
EM22	8-123	23-20#												
EM23	8-130	23-21#												
EM24	8-137	23-22#												
EM25	8-145	23-23#												
EM26	8-153	23-24#												
EM27	8-160	23-25#												
EM3	8-18	23-5#												
EM30	8-167	23-26#												
EM31	8-174	23-27#												
EM32	8-181	23-28#												
EM33	8-188	23-29#												
EM34	8-195	23-30#												
EM35	8-202	23-31#												
EM36	8-209	23-32#												
EM37	8-216	23-33#												
EM4	8-25	23-6#												
EM40	8-223	23-34#												
EM41	8-230	23-35#												
EM42	8-237	23-36#												
EM43	8-244	23-37#												
EM44	8-251	23-38#												
EM45	8-258	23-39#												
EM46	8-265	23-40#												
EM47	8-272	23-41#												
EM5	8-32	23-7#												
EM50	8-279	23-42#												
EM51	8-286	23-43#												
EM52	8-293	23-44#												
EM53	8-300	23-45#												
EM54	8-307	23-46#												
EM55	8-314	23-47#												
EM56	8-321	23-48#												
EM57	8-329	23-49#												
EM6	8-39	23-8#												
EM60	8-336	23-50#												
EM61	8-343	23-51#												
EM62	8-350	23-52#												
EM63	8-357	23-53#												
EM64	8-364	23-54#												
EM65	8-371	23-55#												
EM7	8-46	23-9#												
EMTVEC	4-595#	9-23*	9-23*											
ENTERA	9-31	22-3#												
ERR	4-659#	9->96	9->96	9->96	9-?14	9-?14	9-?14							
ERROR	4-595#													
ERRVEC	4-595#	9-23	9-23*	9-23*	9-25*	9-26*	11-6*	11-7*	11-16*	11-25*	12-1	12-1	12-1*	12-1*
	12-1*	12-1*	12-1*	13-1	13-1*	13-1*								
EXEC	9-58	9-65#	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488	9-545	9-568
	9-620	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996	9-:51	9-:69
	9-;24	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11	9-@76	9-A05
	9-A73	10-1												
FO	4-637#													

F1	4-638#													
F2	4-639#													
F3	4-640#													
F4	4-641#													
FER	4-668#													
FMT16	4-741#													
GNS	5-1	5-1	9-6	9-28	10-1	10-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1
	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1
	21-1	21-1												
GO	4-636#													
GTSWR	9-28	21-1#												
HCE	4-671#													
HCI	4-739#													
HCRC	4-672#													
HT	4-595#	15-1	15-1											
IAE	4-674#													
IBSAVE	13-1	13-1	13-1	13-1	13-1	13-1#	13-1*	13-1*	13-1*					
IE	4-601#													
ILF	4-664#													
ILR	4-665#													
IOTVEC	4-595#	9-23*	9-23*											
IR	4-618#													
IVC	4-732#													
KYBCTL	7-0#	9-73*	9-90*	9-136	9-136*	9-198	9-198*	9-287	9-287*	9-305	9-305*	9-355	9-355*	9-372
	9-372*	9-395	9-395*	9-418	9-418*	9-464	9-464*	9-488	9-488*	9-545	9-545*	9-568	9-568*	9-620
	9-620*	9-636	9-636*	9-661	9-661*	9-681	9-681*	9-717	9-717*	9-736	9-736*	9-805	9-805*	9-842
	9-842*	9-893	9-893*	9-919	9-919*	9-973	9-973*	9-996	9-996*	9-:51	9-:51*	9-:69	9-:69*	9-;24
	9-;24*	9-;84	9-;84*	9-<34	9-<34*	9-<55	9-<55*	9-=03	9-=03*	9-=57	9-=57*	9->22	9->22*	9->44
	9->44*	9-?12	9-?12*	9-?32	9-?32*	9-?86	9-?86*	9-@11	9-@11*	9-@76	9-@76*	9-A05	9-A05*	9-A73
	9-A73*	10-1												
LBC	4-730#													
LBT	4-655#													
LF	4-595#	15-1	15-1	22-7	22-8	22-11	22-12	24-3	24-6	24-9	24-11	24-13	24-15	24-17
	24-21	24-23												
LSC	4-731#													
MAXTN	9-84	25-29#												
MCPE	4-606#	9-443	9-466	9-A46	9-A46	9-A76	9-A76							
MDPE	4-620#													
MOH	4-714#													
MOL	4-657#	9-164	9-164	9-267	9-267	9-289	9-289	9-339	9-357	9-384	9-397	9-443	9-443	9-443
	9-466	9-466	9-466	9-523	9-547	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-647
	9-647	9-663	9-663	9-699	9-699	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-821
	9-872	9-872	9-895	9-895	9-953	9-953	9-975	9-975	9-:30	9-:30	9-:53	9-:53	9-<19	9-<36
	9-<85	9-=05	9-=99	9-=99	9-=99	9->24	9->24	9->24	9-?68	9-?88	9-@56	9-@78	9-A45	9-A45
	9-A75	9-A75												
MUR	4-684#	9-=99	9->24											
MXF	4-621#													
NBA	4-716#													
NED	4-624#	9-149	9-149	9-149	9-149									
NEM	4-623#													
NOATA	8-379#	9-267	9-267	9-289	9-289	9-339	9-339	9-357	9-357	9-384	9-384	9-384	9-384	9-397
	9-397	9-397	9-397	9-443	9-443	9-443	9-443	9-466	9-466	9-466	9-466	9-523	9-523	9-523
	9-523	9-547	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-647
	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-699	9-699	9-699	9-699	9-719	9-719	9-719
	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-821	9-821	9-821	9-821	9-872
	9-872	9-872	9-872	9-872#	9-872#	9-895	9-895	9-895	9-895	9-895#	9-895#	9-897#	9-953	9-953

	9-953	9-953	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53
	9-;05	9-;05	9-;05	9-;05	9-;05#	9-;07#	9-;69	9-;69	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19
	9-<19#	9-<19#	9-<36	9-<36	9-<36	9-<36	9-<36#	9-<36#	9-<85	9-<85	9-<85	9-<85	9-=05	9-=05
	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9-=99#	9->24	9->24	9->24	9->24	9->24#	9->26#	9->96
	9->96	9->96	9->96	9->96#	9->96#	9-?14	9-?14	9-?14	9-?14	9-?14#	9-?14#	9-?68	9-?68	9-?68
	9-?68	9-?88	9-?88	9-?88	9-?88	9-@56	9-@56	9-@56	9-@56	9-@56#	9-@56#	9-@78	9-@78	9-@78
	9-@78	9-@78#	9-@78#	9-A45	9-A45	9-A45	9-A45	9-A75	9-A75	9-A75	9-A75			
NOCLOC	9-59	22-7#												
NOSEIZ	7-0#													
NTRH	9-104	22-12#												
OFD	4-738#													
OM	4-650#	9-267	9-267	9-267	9-289	9-289	9-289	9-339	9-339	9-357	9-357	9-384	9-384	9-397
	9-397	9-443	9-443	9-443	9-466	9-466	9-466	9-523	9-523	9-547	9-547	9-599	9-599	9-599
	9-622	9-622	9-622	9-647	9-647	9-647	9-663	9-663	9-663	9-699	9-699	9-699	9-719	9-719
	9-719	9-791	9-791	9-791	9-791	9-821	9-821	9-872	9-872	9-895	9-895	9-953	9-953	9-953
	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-;05	9-;05	9-;69	9-;69	9-<19
	9-<19	9-<36	9-<36	9-<85	9-<85	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24
	9->24	9->96	9->96	9-?14	9-?14	9-?68	9-?68	9-?88	9-?88	9-@56	9-@56	9-@78	9-@78	9-A45
	9-A45	9-A75	9-A75											
OPE	4-733#													
OPI	4-677#													
OPPRT	7-0#	9-267*	9-289*	9-339*	9-357*	9-443*	9-466*	9-523*	9-547*	9-599*	9-622*	9-647*	9-663*	9-699*
	9-719*	9-791*	9-791*	9-872*	9-895*	9-953*	9-975*	9-:30*	9-:53*	9-=99*	9->24*	9->96*	9-?14*	9-?68*
	9-?88*	9-@56*	9-@78*	9-A45*	9-A75*	24-35	24-40	24-45						
OR	4-619#													
PAR	4-667#													
PAT	4-616#													
PFECH	14-1	14-1#												
PFECH1	14-1	14-1#												
PFECH2	14-1	14-1#												
PFECH3	14-1	14-1#												
PFECH4	14-1	14-1#												
PFTSTN	14-1	14-1#	14-1*											
PGE	4-622#													
PGM	4-654#	9-164	9-164	9-164	9-164	9-164	9-164	9-267	9-267	9-289	9-289	9-339	9-357	9-384
	9-397	9-443	9-443	9-443	9-466	9-466	9-466	9-523	9-547	9-599	9-599	9-599	9-599	9-622
	9-622	9-622	9-622	9-647	9-647	9-663	9-663	9-699	9-699	9-719	9-719	9-791	9-791	9-791
	9-791	9-791	9-791	9-821	9-872	9-872	9-895	9-895	9-953	9-953	9-975	9-975	9-:30	9-:30
	9-:53	9-:53	9-<19	9-<36	9-<85	9-=05	9-=99	9-=99	9-=99	9->24	9->24	9->24	9-?68	9-?88
	9-@56	9-@78	9-A45	9-A45	9-A75	9-A75								
PIP	4-658#	9-267	9-267	9-267	9-289	9-289	9-289	9-339	9-339	9-357	9-357	9-384	9-384	9-397
	9-397	9-443	9-443	9-443	9-466	9-466	9-466	9-523	9-523	9-547	9-547	9-599	9-599	9-599
	9-622	9-622	9-622	9-647	9-647	9-647	9-663	9-663	9-663	9-699	9-699	9-699	9-719	9-719
	9-719	9-791	9-791	9-791	9-791	9-821	9-821	9-872	9-872	9-895	9-895	9-953	9-953	9-953
	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-;05	9-;05	9-;69	9-;69	9-<19
	9-<19	9-<36	9-<36	9-<85	9-<85	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24
	9->24	9->96	9->96	9-?14	9-?14	9-?68	9-?68	9-?88	9-?88	9-@56	9-@56	9-@78	9-@78	9-A45
	9-A45	9-A75	9-A75											
PIRQ	4-595#													
PIRQVE	4-595#													
PORTA	7-0#	9-33*	9-34	9-38	9-41	9-45	9-49	9-52	9-149	9-149	9-156	9-156	9-164	9-164
	9-168	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-289
	9-289	9-289	9-289	9-289	9-289	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339
	9-339	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-384	9-384	9-384	9-384
	9-384	9-384	9-384	9-397	9-397	9-397	9-397	9-443	9-443	9-443	9-443	9-443	9-443	9-443
	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-466	9-466	9-466	9-466	9-466	9-466

	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-523	9-523	9-523	9-523	9-523	9-523	9-523
	9-523	9-523	9-523	9-523	9-523	9-523	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547
	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599
	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-647	9-647	9-647
	9-647	9-647	9-647	9-647	9-647	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-663	9-663
	9-663	9-663	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-719	9-719
	9-719	9-719	9-719	9-719	9-719	9-719	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-821	9-821	9-821	9-821	9-821	9-872	9-872	9-872
	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-895	9-895
	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-953	9-953	9-953
	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-975	9-975	9-975
	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05	9-;05	9-;05	9-;05
	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69
	9-;69	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36	9-<36	9-<36	9-<36	9-<36
	9-<36	9-<36	9-<36	9-<36	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85
	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24
	9->24	9->24	9->24	9->24	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96
	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?68	9-?68	9-?68	9-?68
	9-?68	9-?68	9-?68	9-?88	9-?88	9-?88	9-?88	9-?88	9-?88	9-?88	9-?88	9-a56	9-a56	9-a56
	9-a56	9-a56	9-a56	9-a56	9-a56	9-a78	9-a78	9-a78	9-a78	9-a78	9-a78	9-a78	9-a78	9-a78
	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45
	9-A45	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75
	9-A75	9-A75												
PORTAI	9-44	22-5#												
PORTB	7-0#	9-38*	9-39*	9-40*	9-43*	9-47	9-149	9-149	9-156	9-156	9-164	9-164	9-170	9-267
	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-289
	9-289	9-289	9-289	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-357	9-357
	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-384	9-384	9-384	9-384	9-397	9-397
	9-397	9-397	9-397	9-397	9-397	[illegible]-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443
	9-443	9-443	9-443	9-443	9-[illegible]	[illegible]-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466
	9-466	9-466	9-466	9-466	9-[illegible]	9-523	9-523	9-523	9-523	9-523	9-523	9-523	9-523	9-547
	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-599	9-599
	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-622
	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-647	9-647	9-647	9-647	9-647
	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-663	9-663	9-663	9-663	9-663	9-663	9-663
	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-719	9-719	9-719
	9-719	9-719	9-719	9-719	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-821	9-821	9-821	9-821	9-821	9-872	9-872	9-872	9-872	9-872	9-872
	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-895	9-895	9-895	9-895	9-895	9-895	9-895
	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-953	9-953	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-953	9-953	9-953	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975
	9-975	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69	9-;69	9-;69
	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19
	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36	9-<36	9-<36	9-<36	9-<85	9-<85	9-<85	9-<85	9-<85
	9-<85	9-<85	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=99	9-=99
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->96	9->96	9->96	9->96	9->96
	9->96	9->96	9->96	9->96	9->96	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14
	9-?14	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68	9-?88	9-?88	9-?88	9-?88	9-?88

	9-?88	9-?88	9-@56	9-@56	9-@56	9-@56	9-@56	9-@56	9-@56	9-@56	9-@56	9-@78	9-@78	9-@78
	9-@78	9-@78	9-@78	9-@78	9-@78	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45
	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75
	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75							
PORTBI	9-46	22-6#												
PORTC	7-0#	9-49*	9-50*	9-51*	9-791	9-821								
PR0	4-595#													
PR1	4-595#													
PR2	4-595#													
PR3	4-595#	12-5												
PR4	4-595#													
PR5	4-595#	9-92	9-134	12-9										
PR6	4-595#	9-26												
PR7	4-595#													
PS	4-595	4-595#	9-66*	9-267*	9-267*	9-267*	9-289*	9-289*	9-289*					
PSEL	4-605#													
PSW	4-595#													
PTNBR	7-0#	9-149*	9-149*	9-156*	9-156*	9-164*	9-164*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*
	9-267*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-289*	9-339*	9-339*	9-339*	9-339*	9-339*
	9-339*	9-339*	9-339*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-384*	9-384*	9-384*
	9-384*	9-384*	9-397*	9-397*	9-397*	9-397*	9-397*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*
	9-443*	9-443*	9-443*	9-443*	9-443*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*
	9-466*	9-466*	9-466*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-547*	9-547*	9-547*
	9-547*	9-547*	9-547*	9-547*	9-547*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*
	9-599*	9-599*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-647*
	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*	9-699*
	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*
	9-719*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-821*	9-821*	9-821*	9-821*	9-872*
	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-895*	9-895*	9-895*
	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-953*	9-953*	9-953*	9-953*	9-953*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*
	9-975*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*
	9-;05*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*
	9-<19*	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*
	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*
	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?68*	9-?68*	9-?68*	9-?68*	9-?68*	9-?88*	9-?88*	9-?88*	9-?88*	9-?88*	9-@56*	9-@56*	9-@56*
	9-@56*	9-@56*	9-@56*	9-@78*	9-@78*	9-@78*	9-@78*	9-@78*	9-@78*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*
	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*
	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	24-32	24-34	24-36	24-37	24-38	24-42	
PWRVEC	4-595#													
R6	4-595#	9-23	9-23*	9-23*										
R7	4-595#													
RDCHR	18-1	21-1#												
RDLIN	19-1	21-1#												
RDOCT	9-32	9-79	9-105	21-1#										
RDY	4-602#													
RELERR	7-0#	9-267*	9-267*	9-289*	9-289*	9-339	9-339*	9-339*	9-357	9-357*	9-357*	9-384*	9-384*	9-397*
	9-397*	9-443*	9-443*	9-443*	9-466*	9-466*	9-466*	9-523*	9-523*	9-547*	9-547*	9-599*	9-599*	9-622*
	9-622*	9-647*	9-647*	9-663*	9-663*	9-699*	9-699*	9-719*	9-719*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-821*	9-821*	9-872*	9-872*	9-872*	9-895*	9-895*	9-895*	9-953*	9-953*	9-975*	9-975*	9-:30*
	9-:30*	9-:53*	9-:53*	9-;05	9-;05*	9-;05*	9-;69	9-;69*	9-;69*	9-<19*	9-<19*	9-<36*	9-<36*	9-<85*
	9-<85*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*	9->24*	9->96*	9->96*	9-?14*

	9-?14*	9-?68*	9-?68*	9-?88*	9-?88*	9-@56*	9-@56*	9-@78*	9-@78*	9-A45*	9-A45*	9-A45*	9-A75*	9-A75*
	9-A75*													
RELOK	8-378#	9-267	9-289	9-339	9-357	9-360#	9-384	9-397	9-443	9-466	9-523	9-547	9-599	9-622
	9-647	9-663	9-699	9-719	9-791	9-791	9-821	9-872	9-895	9-953	9-975	9-:30	9-:53	9-;05
	9-;69	9-<19	9-<36	9-<85	9-=05	9-=99	9-=99	9->24	9->24	9->96	9-?14	9-?68	9-?88	9-@56
	9-@78	9-A45	9-A75											
RESREG	21-1#													
RESVEC	4-595#													
RMAS	4-764#	9-791*	9-821*	9-;69	9-;69	9-;69*	9-<19	9-<19	9-<36	9-<36	9-A45*	9-A75*		
RMBA	4-759#													
RMCS1	4-757#	9-339*	9-339*	9-357*	9-357*	9-384*	9-384*	9-384*	9-397*	9-397*	9-397*	9-443	9-443	9-443*
	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-466	9-466	9-466*	9-466*	9-466*
	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-523	9-523	9-523*	9-523*	9-523*	9-523*	9-523*
	9-523*	9-523*	9-523*	9-523*	9-523*	9-547	9-547	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*
	9-547*	9-547*	9-547*	9-599*	9-599*	9-599*	9-599*	9-622*	9-622*	9-622*	9-622*	9-647	9-647*	9-647*
	9-647*	9-647*	9-647*	9-647*	9-647*	9-663	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*	9-699
	9-699	9-699	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-719	9-719	9-719	9-719*	9-719*
	9-719*	9-719*	9-719*	9-719*	9-719*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-791*	9-791*	9-821*	9-821*	9-821*	9-821*	9-821*	9-821*	9-872*	9-872*	9-872*	9-872*	9-872*
	9-872*	9-872*	9-872*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-953*	9-953*	9-953*
	9-953*	9-975*	9-975*	9-975*	9-975*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05*
	9-;05*	9-;69*	9-;69*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<36*	9-<36*	9-<36*	9-<36*
	9-<36*	9-<36*	9-<36*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-=05*	9-=05*	9-=05*
	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9-?14*	9-?14*	9-?14*
	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?68*	9-?68*	9-?68*	9-?68*	9-?68*	9-?68*	9-?88*	9-?88*	9-?88*
	9-?88*	9-?88*	9-?88*	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*	9-@56*	9-@78*	9-@78*	9-@78*	9-@78*
	9-@78*	9-@78*	9-@78*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A46	9-A46	9-A46
	9-A46	9-A46*	9-A46*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A76	9-A76	9-A76
	9-A76	9-A76*	9-A76*											
RMCS2	4-761#	9-138*	9-149	9-149	9-149	9-149	9-149*	9-149*	9-149*	9-149*	9-156*	9-156*	9-164*	9-164*
	9-168*	9-170*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-267*	9-289*	9-289*	9-289*	9-289*
	9-289*	9-289*	9-289*	9-289*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*	9-339*
	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-357*	9-384*	9-384*	9-384*	9-384*
	9-384*	9-397*	9-397*	9-397*	9-397*	9-397*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*
	9-443*	9-443*	9-443*	9-443*	9-443*	9-443*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*
	9-466*	9-466*	9-466*	9-466*	9-466*	9-466*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*	9-523*
	9-523*	9-523*	9-523*	9-523*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*	9-547*
	9-547*	9-547*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*	9-599*
	9-599*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*	9-622*
	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-647*	9-663*	9-663*	9-663*	9-663*
	9-663*	9-663*	9-663*	9-663*	9-663*	9-663*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*	9-699*
	9-699*	9-699*	9-699*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*	9-719*
	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*
	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-791*	9-821*	9-821*	9-821*	9-821*	9-821*
	9-821*	9-821*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*	9-872*
	9-872*	9-872*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*	9-895*
	9-895*	9-895*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*	9-953*
	9-953*	9-953*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*	9-975*
	9-975*	9-975*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*	9-:30*
	9-:30*	9-:30*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*	9-:53*
	9-:53*	9-:53*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*	9-;05*
	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19*	9-<19*	9-<19*
	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<19*	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*	9-<36*
	9-<36*	9-<36*	9-<36*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-<85*	9-=05*	9-=05*
	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=05*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*

	9-=99*	9-=99*	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*	9->24*
	9->24*	9->24*	9->24*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*	9->96*
	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?14*	9-?68*	9-?68*	9-?68*
	9-?68*	9-?68*	9-?68*	9-?68*	9-?68*	9-?88*	9-?88*	9-?88*	9-?88*	9-?88*	9-?88*	9-?88*	9-?88*	9-a56*
	9-a56*	9-a56*	9-a56*	9-a56*	9-a56*	9-a56*	9-a56*	9-a56*	9-a78*	9-a78*	9-a78*	9-a78*	9-a78*	9-a78*
	9-a78*	9-a78*	9-a78*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*	9-A45*
	9-A45*	9-A45*	9-A45*	9-A45*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*
	9-A75*	9-A75*	9-A75*	9-A75*	9-A75*									
RMDA	4-760#	9-267	9-267	9-267*	9-289	9-289	9-289*							
RMDB	4-766#													
RMDC	4-771#	9-267	9-267	9-289	9-289									
RMDS	4-762#	9-149	9-149	9-164	9-164	9-164	9-164	9-164	9-164	9-164	9-164	9-267	9-267	9-267
	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-339	9-339	9-339
	9-339	9-339	9-339	9-339	9-339	9-339	9-339	9-339*	9-357	9-357	9-357	9-357	9-357	9-357
	9-357	9-357	9-357	9-357	9-357*	9-384	9-384	9-384	9-397	9-397	9-397	9-443	9-443	9-443
	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443*	9-443*	9-443*	9-466	9-466	9-466
	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466*	9-466*	9-466*	9-523	9-523	9-523
	9-523	9-523	9-523*	9-523*	9-523*	9-547	9-547	9-547	9-547	9-547	9-547*	9-547*	9-547*	9-599
	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-599*
	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622
	9-622*	9-647	9-647	9-647	9-647	9-647	9-647	9-647*	9-647*	9-663	9-663	9-663	9-663	9-663
	9-663	9-663*	9-663*	9-699	9-699	9-699	9-699	9-699	9-699	9-699*	9-699*	9-719	9-719	9-719
	9-719	9-719	9-719	9-719*	9-719*	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791*	9-791*	9-821	9-821
	9-821	9-821*	9-821*	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872*
	9-872*	9-872*	9-872*	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895*
	9-895*	9-895*	9-895*	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-953	9-953	9-953	9-953*	9-975	9-975	9-975	9-975	9-975	9-975	9-975
	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975*	9-:30	9-:30	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30
	9-:30*	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53*	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05
	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69
	9-;69	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19	9-<19*	9-<19*	9-<19*	9-<36	9-<36	9-<36	9-<36
	9-<36*	9-<36*	9-<36*	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85*	9-<85*	9-=05
	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05*	9-=05*	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9->24
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24
	9->24	9->24	9->24	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96*
	9->96*	9->96*	9->96*	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14*
	9-?14*	9-?14*	9-?14*	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68*	9-?68*	9-?68*	9-?88	9-?88	9-?88
	9-?88	9-?88	9-?88*	9-?88*	9-?88*	9-a56	9-a56	9-a56	9-a56	9-a56	9-a56*	9-a56*	9-a56*	9-a56*
	9-a78	9-a78	9-a78	9-a78	9-a78	9-a78*	9-a78*	9-a78*	9-a78*	9-A45	9-A45	9-A45	9-A45	9-A45
	9-A45	9-A45*	9-A45*	9-A45*	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75*	9-A75*	9-A75*	
RMDT	4-768#	9-156	9-156	9-156	9-156	9-267	9-267	9-289	9-289					
RMEC1	4-774#	9-267	9-267	9-289	9-289									
RMEC2	4-775#	9-267	9-267	9-289	9-289									
RMER1	4-763#	9-267	9-267	9-289	9-289	9-523	9-523	9-523*	9-547	9-547	9-547*	9-953*	9-953*	9-953*
	9-953*	9-975*	9-975*	9-975*	9-975*	9-:30*	9-:30*	9-:30*	9-:30*	9-:53*	9-:53*	9-:53*	9-:53*	9-;05*
	9-;05*	9-;05*	9-;05*	9-;69*	9-;69*	9-;69*	9-;69*	9-<19*	9-<19*	9-<36*	9-<36*	9->96	9->96	9->96*
	9-?14	9-?14	9-?14*	9-A45*	9-A45*	9-A45*	9-A46	9-A46	9-A75*	9-A75*	9-A75*	9-A76	9-A76	
RMER2	4-773#	9-267	9-267	9-289	9-289	9-A45*	9-A45*	9-A45*	9-A46	9-A46	9-A75*	9-A75*	9-A75*	9-A76
	9-A76													
RMLA	4-765#	9-267	9-267	9-289	9-289									
RMMR1	4-767#	9-267	9-267	9-289	9-289	9-=99*	9-=99*	9-=99*	9->24*	9->24*	9->24*			
RMMR2	4-772#													

RMOF	4-770#	9-267	9-267	9-289	9-289									
RMR	4-666#													
RMSN	4-769#	9-169	9-171	9-267	9-267	9-289	9-289							
RMWC	4-758#	9-112												
RQA	4-686#													
RQB	4-685#													
RQSTA	7-0#													
RQSTB	7-0#													
SAVREG	21-1#													
SC	4-608#													
SC0	4-720#													
SC1	4-721#													
SC2	4-722#													
SC3	4-723#													
SC4	4-724#													
SCOPE	4-595#	9-178	9-267	9-289	9-339	9-357	9-384	9-397	9-443	9-466	9-523	9-547	9-599	9-622
	9-647	9-663	9-699	9-719	9-791	9-821	9-872	9-895	9-953	9-975	9-:30	9-:53	9-;05	9-;69
	9-<19	9-<36	9-<85	9-=05	9-=99	9->24	9->96	9-?14	9-?68	9-?88	9-@56	9-@78	9-A45	9-A75
	9-A80													
SEIZPT	7-0#	9-267*	9-289*	9-339*	9-357*	9-384*	9-397*	9-443*	9-466*	9-523*	9-547*	9-599*	9-622*	9-647*
	9-663*	9-699*	9-719*	9-791*	9-791*	9-791*	9-872*	9-895*	9-953*	9-975*	9-:30*	9-:53*	9-<19*	9-<36*
	9-<85*	9-=05*	9-=99*	9->24*	9->96*	9-?14*	9-?68*	9-?88*	9-@56*	9-@78*	9-A45*	9-A75*	24-36	24-37
	24-38	24-39	24-40	24-41	24-45	24-46								
SKI	4-734#													
STACK	4-595#	9-23	9-69	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488	9-545
	9-568	9-620	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996	9-:51
	9-:69	9-;24	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11	9-@76
	9-A05	9-A73												
START	5-1	9-14#	18-1	18-1										
START1	5-3	9-17#												
START2	9-16	9-19#												
STKLMT	4-595#													
STOP	12-1	12-5#												
SW0	4-595#													
SW00	4-595	4-595#												
SW01	4-595	4-595#												
SW02	4-595	4-595#												
SW03	4-595	4-595#												
SW04	4-595	4-595#												
SW05	4-595	4-595#												
SW06	4-595	4-595#												
SW07	4-595	4-595#												
SW08	4-595	4-595#												
SW09	4-595	4-595#												
SW1	4-595#													
SW10	4-595#													
SW11	4-595#													
SW12	4-595#													
SW13	4-595#													
SW14	4-595#													
SW15	4-595#	9-175												
SW2	4-595#													
SW3	4-595#													
SW4	4-595#													
SW5	4-595#													
SW6	4-595#													

SW7	4-595#													
SW8	4-595#													
SW9	4-595#													
SWR	6-0#	9-23	9-23	9-23*	9-23*	9-28	9-175	12-1	12-1	12-1	12-1*	12-1*	12-1*	12-1*
	13-1	13-1	13-1	13-1	18-1	18-1	18-1*							
SWREG	5-1#	9-23	9-28	18-1	18-1	18-1								
TAP	4-715#													
TBITVE	4-595#													
TEST1	9-136	9-136	9-136#											
TEST10	9-418	9-418	9-418#											
TEST11	9-464	9-464	9-464#											
TEST12	9-488	9-488	9-488#											
TEST13	9-545	9-545	9-545#											
TEST14	9-568	9-568	9-568#											
TEST15	9-620	9-620	9-620#											
TEST16	9-636	9-636	9-636#											
TEST17	9-661	9-661	9-661#											
TEST2	9-198	9-198	9-198#											
TEST20	9-681	9-681	9-681#											
TEST21	9-717	9-717	9-717#											
TEST22	9-736	9-736	9-736#											
TEST23	9-805	9-805	9-805#											
TEST24	9-842	9-842	9-842#											
TEST25	9-893	9-893	9-893#											
TEST26	9-919	9-919	9-919#											
TEST27	9-973	9-973	9-973#											
TEST3	9-287	9-287	9-287#											
TEST30	9-996	9-996	9-996#											
TEST31	9-:51	9-:51	9-:51#											
TEST32	9-:69	9-:69	9-:69#											
TEST33	9-;24	9-;24	9-;24#											
TEST34	9-;84	9-;84	9-;84#											
TEST35	9-<34	9-<34	9-<34#											
TEST36	9-<55	9-<55	9-<55#											
TEST37	9-=03	9-=03	9-=03#											
TEST4	9-305	9-305	9-305#											
TEST40	9-=57	9-=57	9-=57#											
TEST41	9->22	9->22	9->22#											
TEST42	9->44	9->44	9->44#											
TEST43	9-?12	9-?12	9-?12#											
TEST44	9-?32	9-?32	9-?32#											
TEST45	9-?86	9-?86	9-?86#											
TEST46	9-@11	9-@11	9-@11#											
TEST47	9-@76	9-@76	9-@76#											
TEST5	9-355	9-355	9-355#											
TEST50	9-A05	9-A05	9-A05#											
TEST51	9-A73	9-A73	9-A73#											
TEST6	9-372	9-372	9-372#											
TEST7	9-395	9-395	9-395#											
TESTNO	9-78	22-8#												
TIME	7-0#	9-267	9-267	9-267*	9-289	9-289	9-289*	9-339*	9-357*	9->96*	9-?14*	9-?68	9-?68*	9-?68*
	9-?88	9-?88*	9-?88*	11-30*	11-32*	24-46								
TIMEA	7-0#	9-56*	9-267	9-267*	9-267*									
TIMEAM	7-0#	9-267*	9-267*	9-?68										
TIMEAP	7-0#	9-56*	9-267*	9-267*	9-?68	9-@56								
TIMEB	7-0#	9-56*	9-289	9-289*	9-289*									

```
TIMEBM   7-0#     9-289*   9-289*   9-?88
TIMEBP   7-0#     9-56*    9-289*   9-289*   9-?88    9-@78
TIMES    7-0#     9-?68    9-?68    9-?68*   9-?88    9-?88    9-?88*
TKVEC    4-595#  18-1*    18-1*
TOLER    9-267    9-289   11-42#
TPVEC    4-595#
TRAPVE   4-595#   9-23*    9-23*
TRE      4-607#   9-443    9-466    9-A46    9-A46    9-A76    9-A76
TRTVEC   4-595#
TST1     9-136#  25-5
TST10    9-418#  25-8
TST11    9-464#  25-8
TST12    9-488#  25-8
TST13    9-545#  25-8
TST14    9-568#  25-8
TST15    9-620#  25-8
TST16    9-636#  25-8
TST17    9-661#  25-8
TST1AA   9-82     9-133#  10-1
TST2     9-198#  25-8
TST20    9-681#  25-8
TST21    9-717#  25-8
TST22    9-736#  25-8
TST23    9-805#  25-8
TST24    9-842#  25-8
TST25    9-893#  25-11
TST26    9-919#  25-11
TST27    9-973#  25-11
TST3     9-287#  25-8
TST30    9-996#  25-11
TST31    9-:51#  25-11
TST32    9-:69#  25-11
TST33    9-;24#  25-11
TST34    9-;84#  25-11
TST35    9-<34#  25-11
TST36    9-<55#  25-11
TST37    9-=03#  25-14
TST4     9-305#  25-8
TST40    9-=57#  25-14
TST41    9->22#  25-14
TST42    9->44#  25-14
TST43    9-?12#  25-14
TST44    9-?32#  25-14
TST45    9-?86#  25-14
TST46    9-@11#  25-14
TST47    9-@76#
TST5     9-355#  25-8
TST50    9-A05#
TST50B   9-A45    9-A45    9-A46#
TST51    9-A46    9-A73#
TST51B   9-A75    9-A75    9-A76#
TST52    9-A76    9-A80#
TST6     9-372#  25-8
TST7     9-395#  25-8
TSTADR   9-93    25-5#
TSTERR  22-10#
```

TSTNUM	7-0#	13-1*	24-32	24-33	24-34	24-35	24-36	24-37	24-38	24-39	24-40	24-41	24-42	24-43
	24-44	24-45	24-46	24-48										
TYPDS	10-1	10-1	14-1	21-1#										
TYPE	9-6	9-28	9-31	9-36	9-44	9-46	9-48	9-59	9-67	9-78	9-86	9-100	9-103	9-104
	10-1	10-1	10-1	13-1	13-1	14-1	14-1	14-1	14-1	14-1	14-1	14-1	15-1	16-1
	17-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1	18-1
	18-1	18-1	18-1	18-1	18-1	19-1	19-1	21-1#						
TYPOC	9-8	9-102	14-1	14-1	18-1	21-1#								
TYPON	21-1#													
TYPOS	9-45	9-47	21-1#											
U0	4-612#													
U1	4-613#													
U3	4-614#													
UNS	4-678#													
UPE	4-625#													
VV	4-651#	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-339	9-339	9-339	9-339	9-339
	9-339	9-339	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-384	9-384	9-384	9-397	9-397
	9-397	9-443	9-443	9-443	9-443	9-443	9-466	9-466	9-466	9-466	9-466	9-523	9-523	9-523
	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-622
	9-622	9-647	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-699	9-699	9-699	9-699	9-719
	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-821
	9-821	9-821	9-872	9-872	9-872	9-872	9-895	9-895	9-895	9-895	9-953	9-953	9-953	9-953
	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05
	9-;69	9-;69	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36	9-<85	9-<85	9-<85	9-=05	9-=05	9-=05
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99	9-=99	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24
	9->24	9->24	9->24	9->24	9->96	9->96	9-?14	9-?14	9-?68	9-?68	9-?68	9-?88	9-?88	9-?88
	9-@56	9-@56	9-@56	9-@78	9-@78	9-@78	9-A45	9-A45	9-A45	9-A45	9-A75	9-A75	9-A75	9-A75
VVSET	8-377#	9-267	9-267	9-267	9-267	9-267	9-267	9-289	9-289	9-289	9-289	9-289	9-289	9-339
	9-339	9-339	9-339	9-339	9-357	9-357	9-357	9-357	9-357	9-359#	9-384	9-384	9-384	9-384
	9-384	9-384	9-397	9-397	9-397	9-397	9-397	9-397	9-443	9-443	9-443	9-443	9-443	9-443
	9-443	9-443	9-443	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-523	9-523
	9-523	9-523	9-523	9-523	9-523	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-599	9-599
	9-599	9-599	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622
	9-647	9-647	9-647	9-647	9-647	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-663	9-663
	9-663	9-663	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-699	9-719	9-719	9-719	9-719
	9-719	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-821	9-821	9-821	9-821	9-821	9-821	9-872	9-872
	9-872	9-872	9-872	9-872	9-872	9-872	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-895
	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-953	9-975	9-975	9-975	9-975	9-975	9-975
	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69	9-;69	9-;69	9-;69
	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36	9-<36	9-<36
	9-<36	9-<36	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-=05	9-=05	9-=05	9-=05	9-=05
	9-=05	9-=05	9-=07#	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9->24
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->27#	9->96	9->96	9->96	9->96
	9->96	9->96	9-?14	9-?14	9-?14	9-?14	9-?14	9-?14	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68
	9-?88	9-?88	9-?88	9-?88	9-?88	9-?88	9-@56	9-@56	9-@56	9-@56	9-@56	9-@56	9-@78	9-@78
	9-@78	9-@78	9-@78	9-@78	9-A75	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A75	9-A75
	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75								
WATCH	7-0#	9-267	9-267*	9-289	9-289*	9-339	9-339*	9-357	9-357*	9->96	9->96*	9-?14	9-?14*	9-?68
	9-?68	9-?68*	9-?68*	9-?88	9-?88	9-?88*	9-?88*	9-@56	9-@56*	9-@78	9-@78*	11-33	11-35*	11-37*
WCE	4-626#													
WCF	4-669#													
WLE	4-675#													
WRL	4-656#	9-267	9-267	9-289	9-289	9-339	9-339	9-357	9-357	9-384	9-384	9-397	9-397	9-443

9-443	9-466	9-466	9-523	9-523	9-547	9-547	9-599	9-599	9-622	9-622	9-647	9-647	9-663
9-663	9-699	9-699	9-719	9-719	9-791	9-791	9-791	9-791	9-821	9-821	9-872	9-872	9-895
9-895	9-953	9-953	9-975	9-975	9-:30	9-:30	9-:53	9-:53	9-;05	9-;05	9-;69	9-;69	9-<19
9-<19	9-<36	9-<36	9-<85	9-<85	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9->24	9->24	9->24
9->24	9->96	9->96	9-?14	9-?14	9-?68	9-?68	9-?88	9-?88	9-@56	9-@56	9-@78	9-@78	9-A45
9-A45	9-A75	9-A75											

$$CMRE	5-46#	6-0												
$$CMTM	5-46#	6-0	6-0	6-0	6-0	6-0								
$$ESCA	4-595#													
$$NEWT	4-595#	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488	9-545	9-568	9-620
	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996	9-:51	9-:69	9-;24
	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11	9-@76	9-A05	9-A73
	9-A80													
$$SET	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1#	
$$SKIP	4-595#													
.$ACT1	4-584#	5-5												
.$CATC	4-585#	5-1												
.$CMTA	4-585#	5-46												
.$EOP	4-585#	10-1												
.$ERRO	4-585#	13-1												
.$ERRT	4-584#	14-1												
.$RDOC	4-586#	19-1												
.$READ	4-586#	18-1												
.$SAVE	4-586#	20-1												
.$SCOP	4-585#	12-1												
.$TRAP	4-586#	21-1												
.$TYPD	4-586#	17-1												
.$TYPE	4-585#	15-1												
.$TYPO	4-586#	16-1												
.EQUAT	4-584#	4-595												
.HEADE	4-584#	4-592												
.SETUP	4-584#	4-777												
.SWRHI	4-584#	4-593												
.SWRLO	4-584#	4-593#												
AA	9-200#	9-267	9-289											
ADDO	9-;26#	9-;69												
ADD1	9-998#	9-:30	9-:53											
ADD4	9-=59#	9-=99	9->24											
CC	9-307#	9-339	9-357											
CC1	9-374#	9-384	9-397											
CHECK	4-123#	9-149	9-149	9-156	9-156	9-164	9-164	9-164	9-164	9-339	9-339	9-339	9-357	9-357
	9-357	9-443	9-466	9-523	9-523	9-523	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-622
	9-622	9-622	9-622	9-699	9-719	9-791	9-791	9-872	9-872	9-895	9-895	9-953	9-953	9-953
	9-953	9-975	9-975	9-975	9-975	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53	9-;05
	9-;05	9-;05	9-;05	9-;69	9-;69	9-<85	9-<85	9-=05	9-=05	9-=99	9-=99	9-=99	9-=99	9-=99
	9-=99	9->24	9->24	9->24	9->24	9->24	9->24	9->96	9->96	9->96	9->96	9-?14	9-?14	9-?14
	9-?14	9-@56	9-@78	9-A46	9-A46	9-A76	9-A76							
CLRATA	4-415#	9-443	9-466	9-523	9-547	9-647	9-663	9-699	9-719	9-791	9-821	9-872	9-895	9-<19
	9-<36	9-<85	9-=05	9->96	9->96	9-?14	9-?14	9-?68	9-?88	9-@56	9-@78	9-A45	9-A75	
COMMEN	4-595#													
DD	9-638#	9-647	9-663											
EE	9-844#	9-872	9-895											
ENDCOM	4-595#													
ERROR	4-595#	9-114	9-149	9-149	9-156	9-156	9-164	9-164	9-164	9-164	9-174	9-267	9-267	9-267
	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267	9-267
	9-267	9-267	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-289
	9-289	9-289	9-289	9-289	9-289	9-289	9-289	9-339	9-339	9-339	9-339	9-339	9-339	9-339
	9-339	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-357	9-384	9-384	9-384	9-384	9-397
	9-397	9-397	9-397	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-443	9-466
	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-523	9-523	9-523	9-523	9-523
	9-523	9-523	9-547	9-547	9-547	9-547	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-599
	9-599	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622	9-622

	9-622	9-647	9-647	9-647	9-647	9-647	9-647	9-663	9-663	9-663	9-663	9-663	9-663	9-699
	9-699	9-699	9-699	9-699	9-699	9-699	9-719	9-719	9-719	9-719	9-719	9-719	9-719	9-791
	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-821	9-821	9-821	9-821	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-895
	9-895	9-895	9-895	9-895	9-895	9-895	9-895	9-953	9-953	9-953	9-953	9-953	9-953	9-953
	9-953	9-953	9-953	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-975	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53	9-:53
	9-:53	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;05	9-;69
	9-;69	9-;69	9-;69	9-;69	9-;69	9-;69	9-<19	9-<19	9-<19	9-<19	9-<19	9-<36	9-<36	9-<36
	9-<36	9-<36	9-<85	9-<85	9-<85	9-<85	9-<85	9-<85	9-=05	9-=05	9-=05	9-=05	9-=05	9-=05
	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99	9-=99
	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24	9->24
	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9->96	9-?14	9-?14	9-?14	9-?14	9-?14
	9-?14	9-?14	9-?14	9-?14	9-?68	9-?68	9-?68	9-?68	9-?68	9-?68	9-?88	9-?88	9-?88	9-?88
	9-?88	9-?88	9-@56	9-@56	9-@56	9-@56	9-@56	9-@78	9-@78	9-@78	9-@78	9-@78	9-A45	9-A45
	9-A45	9-A45	9-A45	9-A45	9-A46	9-A46	9-A46	9-A46	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75
	9-A76	9-A76	9-A76	9-A76	12-1									
ESCAPE	4-595#													
FF	9-683#	9-699	9-719											
FF1	9-420#	9-443	9-466											
GETPRI	4-595#													
GETSWR	4-595#	9-28	9-28#											
GG	9-738#	9-791												
HH	9-807#	9-821												
II	9->46#	9->96	9-?14											
JJ	9-490#	9-523	9-547											
KK	9-?34#	9-?68	9-?88											
LL	9-A07#	9-A45	9-A75											
LL1	9-A34#	9-A46	9-A76											
MM	9-<57#	9-<85	9-=05											
MM0	9-:71#	9-;05												
MM1	9-921#	9-953	9-@75											
MORETA	5-7#	6-0												
MSG	9-120#	9-136	9-180#	9-198	9-269#	9-287	9-291#	9-305	9-341#	9-355	9-363#	9-372	9-386#	9-395
	9-399#	9-418	9-445#	9-464	9-468#	9-488	9-525#	9-545	9-549#	9-568	9-601#	9-620	9-624#	9-636
	9-649#	9-661	9-665#	9-681	9-701#	9-717	9-721#	9-736	9-793#	9-805	9-823#	9-842	9-874#	9-893
	9-900#	9-919	9-955#	9-973	9-977#	9-996	9-:32#	9-:51	9-:55#	9-:69	9-;10#	9-;24	9-;71#	9-;84
	9-<21#	9-<34	9-<38#	9-<55	9-<87#	9-=03	9-=10#	9-=20	9-=23#	9-=33	9-=36#	9-=57	9->01#	9->22
	9->30#	9->44	9->98#	9-?12	9-?16#	9-?32	9-?70#	9-?86	9-?90#	9-@11	9-@58#	9-@76	9-@80#	9-A05
	9-A48#	9-A73												
MULT	4-595#													
NEUTRA	4-269#	9-267	9-289	9-339	9-357	9-384	9-397	9-443	9-466	9-523	9-547	9-599	9-622	9-647
	9-663	9-699	9-719	9-791	9-791	9-821	9-872	9-895	9-953	9-975	9-:30	9-:53	9-;05	9-;69
	9-<19	9-<36	9-<85	9-=05	9-=99	9-=99	9->24	9->24	9->96	9-?14	9-?68	9-?88	9-@56	9-@78
	9-A45	9-A75												
NEWTST	4-595#	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488	9-545	9-568	9-620
	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996	9-:51	9-:69	9-;24
	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11	9-@76	9-A05	9-A73
	9-A80													
NN	9-@13#	9-@56	9-@78											
NWTST	4-547#	9-136	9-198	9-287	9-305	9-355	9-372	9-395	9-418	9-464	9-488	9-545	9-568	9-620
	9-636	9-661	9-681	9-717	9-736	9-805	9-842	9-893	9-919	9-973	9-996	9-:51	9-:69	9-;24
	9-;84	9-<34	9-<55	9-=03	9-=57	9->22	9->44	9-?12	9-?32	9-?86	9-@11	9-@76	9-A05	9-A73
OO	9-570#	9-599	9-622											
POP	4-595#	17-1	19-1	20-1										
PUSH	4-595#	17-1	19-1	20-1										

RELEAS	4-197#	9-339	9-357	9-443	9-443	9-466	9-466	9-523	9-547	9-599	9-622	9-647	9-663	9-699
	9-719	9-791	9-791	9-791	9-791	9-872	9-872	9-895	9-895	9-953	9-975	9-:30	9-:53	9-<19
	9-<36	9-<85	9-=05	9-=99	9-=99	9->24	9->24	9-A45	9-A45	9-A75	9-A75			
REPORT	4-595#													
RR	9-:86#	9-<19	9-<36											
SEIZE	4-32#	9-267	9-289	9-339	9-357	9-443	9-466	9-523	9-547	9-599	9-622	9-647	9-663	9-699
	9-719	9-872	9-895	9-953	9-975	9-:30	9-:53	9-=99	9->24	9->96	9-?14	9-?68	9-?88	9-@56
	9-@78	9-A45	9-A75											
SELECT	4-244#	9-149	9-149	9-156	9-156	9-164	9-164	9-267	9-267	9-267	9-289	9-289	9-289	9-339
	9-339	9-339	9-339	9-357	9-357	9-357	9-357	9-384	9-397	9-443	9-443	9-443	9-443	9-443
	9-443	9-443	9-466	9-466	9-466	9-466	9-466	9-466	9-466	9-523	9-523	9-523	9-523	9-547
	9-547	9-547	9-547	9-599	9-599	9-599	9-599	9-599	9-599	9-599	9-622	9-622	9-622	9-622
	9-622	9-622	9-622	9-647	9-647	9-647	9-663	9-663	9-663	9-699	9-699	9-699	9-699	9-719
	9-719	9-719	9-719	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791	9-791
	9-791	9-872	9-872	9-872	9-872	9-872	9-872	9-872	9-895	9-895	9-895	9-895	9-895	9-895
	9-895	9-953	9-953	9-953	9-953	9-953	9-953	9-975	9-975	9-975	9-975	9-975	9-975	9-:30
	9-:30	9-:30	9-:30	9-:30	9-:30	9-:53	9-:53	9-:53	9-:53	9-:53	9-:53	9-;05	9-;05	9-;05
	9-;05	9-;69	9-;69	9-;69	9-;69	9-<19	9-<19	9-<36	9-<36	9-<85	9-<85	9-<85	9-=05	9-=05
	9-=05	9-=99	9-=99	9-=99	9->24	9->24	9->24	9->96	9->96	9->96	9-?14	9-?14	9-?14	9-?68
	9-?88	9-@56	9-@56	9-@78	9-@78	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A45	9-A75
	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75	9-A75							
SETATA	4-448#	9-953	9-975	9-:30	9-:53	9-;05	9-;69							
SETPRI	4-595#	9-92	9-134	12-5	12-9	18-1								
SETTRA	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1	21-1#	
SETUP	4-595#	9-23												
SKIP	4-595#													
SLASH	4-595#													
STARS	4-595#	5-5	6-0	6-0	9-136	9-136	9-198	9-198	9-287	9-287	9-305	9-305	9-355	9-355
	9-372	9-372	9-395	9-395	9-418	9-418	9-464	9-464	9-488	9-488	9-545	9-545	9-568	9-568
	9-620	9-620	9-636	9-636	9-661	9-661	9-681	9-681	9-717	9-717	9-736	9-736	9-805	9-805
	9-842	9-842	9-893	9-893	9-919	9-919	9-973	9-973	9-996	9-996	9-:51	9-:51	9-:69	9-:69
	9-:24	9-:24	9-;84	9-;84	9-<34	9-<34	9-<55	9-<55	9-=03	9-=03	9-=19	9-=21	9-=32	9-=34
	9-=57	9-=57	9->22	9->22	9->44	9->44	9-?12	9-?12	9-?32	9-?32	9-?86	9-?86	9-@11	9-@11
	9-@76	9-@76	9-A05	9-A05	9-A73	9-A73	9-A78	9-A80	10-1	12-1	13-1	14-1	15-1	16-1
	17-1	18-1	18-1	18-1	18-1	18-1	19-1	20-1	21-1					
SWITCH	4-371#	9-443	9-466	9-791	9-791	9-872	9-895	9-A45	9-A75					
SWRSU	4-595#	9-23	9-23#											
TIMER	4-573#	9-267	9-289	9-339	9-357	9->96	9-?14	9-?68	9-?68	9-?88	9-?88			
TRMTRP	21-1#													
TSTREQ	4-481#													
TYPBIN	4-595#													
TYPDEC	4-595#	10-1	10-1	14-1										
TYPNAM	4-595#	9-28												
TYPNUM	4-595#													
TYPOCS	4-595#	9-45	9-47											
TYPOCT	4-595#	14-1	14-1	18-1										
TYPTXT	4-595#	9-6	10-1	10-1										
USEREQ	9-A81#	10-1												