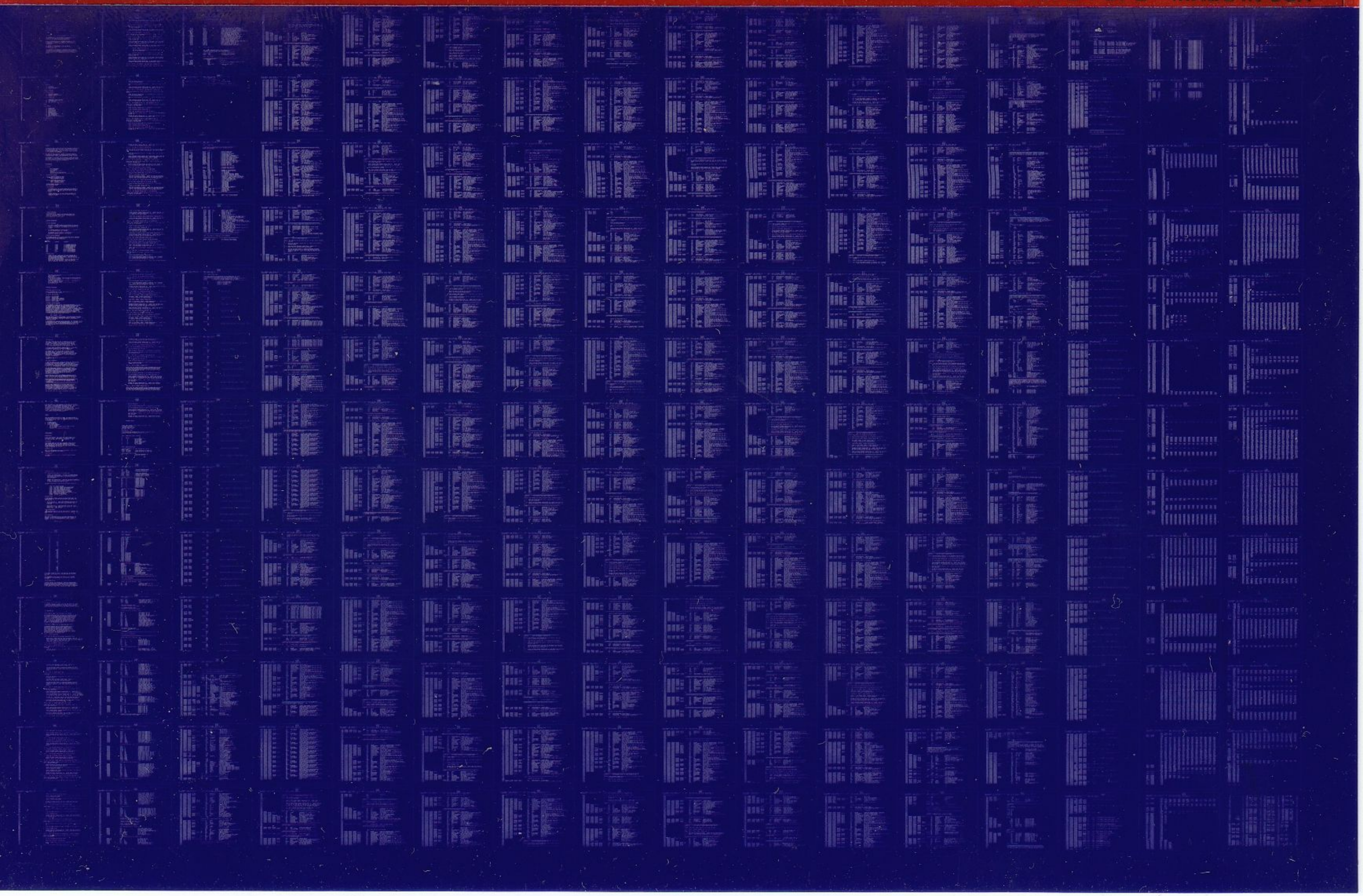
RP04/05/06

DUAL CONTROLLER LOGIC
MD-11-DZRJE-A

EP-DZRJE-A-DL-A

FICHE 1 OF 2

NOV 1976
digital
MADE IN USA



RP04/05/06

DUAL CONTROLLER LOGIC
MD-11-DZRJE-A

EP-DZRJE-A-DL-A

FICHE 2 OF 2

NOV 1976
digital
MADE IN USA

```
.REM @
```

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

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

PRODUCT NAME: RP04/5/6 DUAL CONTROLLER LOGIC TEST - PART 1

DATE CREATED: MARCH 29, 1976

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: C. HESS

CONTENTS

1. ABSTRACT

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

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 PRELIMINARY PROGRAMS

RP04/5/6 DISKLESS CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJG)
PART 2 (MAINDEC-11-DZRJH)

RP04/5/6 FUNCTIONAL CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJI)
PART 2 (MAINDEC-11-DZRJJ)

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

2.3 OTHER PROGRAMS

A. THE OPERATION OF THE UNLOAD COMMAND AND THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH ARE TESTED BY THE RP04/5/6 DUAL CONTROLLER LOGIC TEST, PART 2 (MAINDEC-11-DZRJF).

B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RP04/5/6 MULTIDRIVE EXERCISER PROGRAM (MAINDEC-11-DZRJD).

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 (DCL) ADDRESS.

C. THE PROGRAM CAN BE STARTED AT LOCATION 210 (8) TO ALLOW THE ADDRESS OF THE RH11 OR RH70 TO BE CHANGED.

4.2 UNIBUS & VECTOR ADDRESSES

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

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

4.3 OPERATOR ACTION

A. CONNECT THE DUAL CONTROLLER TEST CABLE BETWEEN BUS A & BUS B ON THE DRIVE BEING TESTED. (SEE SECTION 5.4)
B. LOAD THE PROGRAM INTO MEMORY IN THE PROCESSOR CONTROLLING THE MASSBUS USED FOR TESTING.
C. SWITCH THE 'CONTROLLER SELECT' SWITCH ON THE DRIVE TO BE TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
D. LOAD THE 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. (THIS MUST BE THE NUMBER DISPLAYED BY THE DRIVE, IF AN RP04, OR THE NUMBER OF THE ADDRESS PLUG IF THE DRIVE IS AN RP05/6).

G. ENTER THE NUMBER OF THE TEST TO BE RUN. ('CARRIAGE RETURN' OR '0' WILL RUN ALL TESTS.)

H. THE PROGRAM MAY BE STOPPED AT ANY TIME AND RESTARTED FROM LOCATION 204.

5. OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

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

THE SWITCH SETTINGS ARE:

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

5.2 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RP04/5/6 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

```
                        5.3     TEST SELECTION

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

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

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

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

                        5.4     TEST CABLE CONNECTION

                                TO TEST THE RP04/5/6 DUAL CONTROLLER OPTION WITH THIS PROGRAM,
                                A SPECIAL TEST CABLE MUST BE USED.  (THE TEST CABLE IS
                                P/N 7010507-02).  THE TEST CABLE CONNECTS MASSBUS A & MASSBUS
                                B TOGETHER AT THE DRIVE BEING TESTED AND IS CONSTRUCTED SO
                                THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

                                WITH THE DRIVE CABLE CONNECTED TO THE RP04 UNDER TEST,
                                THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS:  EACH PORT
                                OF THE DRIVE WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS.
                                THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S
                                ADDRESS (THE ADDRESS SELECTED BY THE SWITCHES ON THE
                                'DP' BOARD - MODULE M7775 FOR RP04'S, OR BY THE
                                ADDRESS PLUG FOR RP05/6'S.)

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

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

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

                                WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION
```

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

6. ERRORS

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

A. AN ERROR MESSAGE
B. A DATA HEADER LINE
C. A DATA LINE CONTAINING:
    1. THE TEST NUMBER
    2. THE PC (PROGRAM COUNTER VALUE) WHERE THE ERROR
        CALL WAS MADE
    3. CONTENTS OF THE 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
CONTROLLER OPTION, AND DOES NOT TEST THE UNLOAD COMMAND
OR THE OPERATION OF THE CONTROLLER SELECT SWITCH ON THE DRIVE.
(REFER TO PARAGRAPH 2.2 & 2.3)

7.3 EXECUTION TIME

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

7.4 STACK POINTER

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

7.5 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

A. 'SCOPE' (IOT INSTRUCTION). THIS CALL IS PLACED BETWEEN EACH TEST IN THE INSTRUCTION. THIS ROUTINE ESTABLISHES THE ITERATION COUNT AND THE LOOP ON TEST AND LOOP ON ERROR ADDRESSES.

B. 'ERROR' (EMT INSTRUCTION). THIS CALL IS USED TO REPORT ALL ERRORS. THE CALL IS FOLLOWED BY A NUMBER WHICH IDENTIFIES THE ERROR MESSAGE WHICH WILL BE TYPED.

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

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

7.6 REQUIRED TESTS

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

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

B. TEST 4 AND TEST 5. THESE TESTS SET 'VV-A' AND 'VV-B' RESPECTIVLY. THESE TESTS MUST BE PERFORMED AT LEASE ONCE BEFORE TESTS 6 - 46 ARE RUN.

7.7 DISK SURFACE USAGE

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

7.8 TEST ITERATIONS

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

ITERATION COUNT

TEST NO.	(IN DECIMAL)
01	1
02	10
03	10
04	1
05	1
06	4000
07	4000
10	100
11	100
12	4000
13	4
14	4
15	4
16	4000
17	4000
20	4000
21	4000
22	4000
23	4000
24	4000
25	4000
26	4000
27	4000
30	4000
31	4
32	4
33	4000
34	4000
35	4
36	4
37	4
40	4
41	4000
42	4000
43	4000
44	4000
45	20
46	20

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

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

7.9 LOOP ON ERROR OPTION

IF SW<09> IS SET, THE PROGRAM WILL LOOP ON A FAILING TEST UNTIL EITHER THE SWITCH IS RESET OR THE ERROR STOPS OCCURING. BECAUSE THE PROGRAM MUST RESET THE RP04 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR

WAS DETECTED.

7.10 SPECIAL M7775 'DP' BOARD TESTS

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

8. TEST DESCRIPTIONS

8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

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

8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

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

8.3 TEST 1 - DRIVE ACCESS TEST

VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS

A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RP04/5/6, THAT THE DRIVE IS ONLINE (RPDS1 HAS 'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.

B. THE TEST IS REPEATED THROUGH BOTH PORTS.

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

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

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

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

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

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

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

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

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

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

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

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

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

                        B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
                            'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET

                            FOR PORT 'B'.  (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
                            'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)

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

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

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

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

                        B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
                            'VV' BIT FOR PORT 'B' WAS SET.

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

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

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

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

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

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

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

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

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

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

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

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

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

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

A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.

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

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

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

A. SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.

VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

C. ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.

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

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

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

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

B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

C. ISSUE A MASSBUS CLEAR.

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

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

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

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

B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

C. ISSUE A MASSBUS CLEAR.

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

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

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

A. SET THE ATTENTION BITS FOR BOTH PORTS.

B. VERIFY THAT THE DRIVE IS IN NEUTRAL.

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

8.17 TEST 17 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.

A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'A'; VERIFY THAT THE DRIVE IS SEIZED.

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

8.18 TEST 20 - TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.

A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'B'; VERIFY THAT THE DRIVE IS SEIZED.

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

8.19 TEST 21 - TEST 'PORT REQUEST' FROM PORT 'A'

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

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

B. WRITE 0'S INOT RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.

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

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

8.20 TEST [illegible] - TEST PORT REQUEST FROM PORT 'B'

[illegible] WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
[illegible]IZED BY THE OTHER PORT.

A. [illegible]HE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

B. WRITE 0'S INTO RPDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.

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

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

8.21 TEST 23 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT

REQUEST'.

A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

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

8.22 TEST 24 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'

VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT REQUEST'.

A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.

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

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

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

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

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

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

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

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

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

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

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

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

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

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

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

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

8.25 TEST 27 - TEST SEIZE BY WRITING ATTENTION BIT

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

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

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

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

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

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

B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.

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

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.

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

8.28 TEST 32 - TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.

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

8.29 TEST 33 - TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

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

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.

C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.

D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.

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

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

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

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.

C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.

D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.

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

8.31 TEST 35 - TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

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

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.

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

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

8.33 TEST 37 - TEST PORT 'A' ATTENTION AFTER A COMMAND

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

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

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

C. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED

TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

8.34 TEST 40 - TEST PORT 'B' ATTENTION AFTER A COMMAND

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

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

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

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

8.35 TEST 41 - TEST PORT INTERACTION FROM PORT 'A'

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.

C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.

D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.

E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.

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

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

8.36 TEST 42 - TEST PORT INTERACTION FROM PORT 'B'

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

B. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.

C. READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.

D. CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.

E. WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.

F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS

SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.

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

8.37 TEST 43 - TEST PORT 'A' ALTERNATE ATTENTION BIT PATH

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

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

B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

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

8.38 TEST 44 - TEST PORT 'B' ALTERNATE ATTENTION BIT PATH

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

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

B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

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

8.39 TEST 45 - TEST NO TIMEOUT THROUGH PORT 'A'

VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES. THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST BE SELECTED SEPARATELY.

A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.

B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.

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

D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT BEEN RELEASED.

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

8.40 TEST 46 - TEST NO TIMEOUT THROUGH PORT 'B'

VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES. THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST

```
                                BE SELECTED SEPARATELY.

                                A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

                                B.  SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.

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

                                D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
                                    BEEN RELEASED.

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


                        9.      PROGRAM LISTING
                                ---------------
                                ∂


                        .TITLE  MD-11-DZRJE-A, DUAL CONTRLR LOGIC TEST - PART 1
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY C. HESS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*

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

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
```

```
1097    000015          CR=     15              ;;CODE FOR CARRIAGE RETURN
1098    000200          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
1099    177776          PS=     177776          ;;PROCESSOR STATUS WORD
1100                    .EQUIV  PS,PSW
1101    177774          STKLMT= 177774          ;;STACK LIMIT REGISTER
1102    177772          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
1103    177570          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
1104    177570          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
1105
1106                    ;*GENERAL PURPOSE REGISTER DEFINITIONS
1107    000000          R0=     %0              ;;GENERAL REGISTER
1108    000001          R1=     %1              ;;GENERAL REGISTER
1109    000002          R2=     %2              ;;GENERAL REGISTER
1110    000003          R3=     %3              ;;GENERAL REGISTER
1111    000004          R4=     %4              ;;GENERAL REGISTER
1112    000005          R5=     %5              ;;GENERAL REGISTER
1113    000006          R6=     %6              ;;GENERAL REGISTER
1114    000007          R7=     %7              ;;GENERAL REGISTER
1115    000006          SP=     %6              ;;STACK POINTER
1116    000007          PC=     %7              ;;PROGRAM COUNTER
1117
1118                    ;*PRIORITY LEVEL DEFINITIONS
1119    000000          PR0=    0               ;;PRIORITY LEVEL 0
1120    000040          PR1=    40              ;;PRIORITY LEVEL 1
1121    000100          PR2=    100             ;;PRIORITY LEVEL 2
1122    000140          PR3=    140             ;;PRIORITY LEVEL 3
1123    000200          PR4=    200             ;;PRIORITY LEVEL 4
1124    000240          PR5=    240             ;;PRIORITY LEVEL 5
1125    000300          PR6=    300             ;;PRIORITY LEVEL 6
1126    000340          PR7=    340             ;;PRIORITY LEVEL 7
1127
1128                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1129    100000          SW15=   100000
1130    040000          SW14=   40000
1131    020000          SW13=   20000
1132    010000          SW12=   10000
1133    004000          SW11=   4000
1134    002000          SW10=   2000
1135    001000          SW09=   1000
1136    000400          SW08=   400
1137    000200          SW07=   200
1138    000100          SW06=   100
1139    000040          SW05=   40
1140    000020          SW04=   20
1141    000010          SW03=   10
1142    000004          SW02=   4
1143    000002          SW01=   2
1144    000001          SW00=   1
1145                    .EQUIV  SW09,SW9
1146                    .EQUIV  SW08,SW8
1147                    .EQUIV  SW07,SW7
1148                    .EQUIV  SW06,SW6
1149                    .EQUIV  SW05,SW5
1150                    .EQUIV  SW04,SW4
1151                    .EQUIV  SW03,SW3
1152                    .EQUIV  SW02,SW2
```

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

```
                                RH11 REGISTERS

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)

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

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

                        ;CONTROL AND STATUS REGISTER 2 (RPCS2)

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

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


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

                        .SBTTL  RP04/5/6 REGISTERS

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

                        ;CONTROL AND STATUS 1 REGISTER. (#00)

000001                  GO=     1               ;GO BIT (BIT #0)
000002                  F1=     2               ;FUNCTION CODE BIT #1
000004                  F2=     4               ;FUNCTION CODE BIT #2
000010                  F3=     10              ;FUNCTION CODE BIT #3
000020                  F4=     20              ;FUNCTION CODE BIT #4
000040                  F5=     40              ;FUNCTION CODE BIT #5
004000                  DVA=    4000            ;DEVICE AVAILABLE (BIT #11)

                        ;DRIVE STATUS REGISTER (RPDS1) (#01)

                        ;DF5=   1               DRIVE FORWARD 5"/SEC.  (BIT #0)
000002                  DFF20=  2               ;DRIVE FORWARD 20"/SEC.  (BIT #1)
000004                  DIGB=   4               ;DRIVE TO INNER GUARD BAND (BIT #2)
```

```
1265          000010                  GRV=    10              ;GO REVERSE (BIT #3)
1266          000020                  DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
1267          000040                  DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
1268          000100                  VV=     100             ;VOLUME VALID (BIT #6)
1269          000200                  DRY=    200             ;DRIVE READY (BIT #7)
1270          000400                  DPR=    400             ;DRIVE PRESENT (BIT #8)
1271          001000                  PGM=    1000            ;PROGRAMABLE (BIT #9)
1272          002000                  LST=    2000            ;LAST SECTOR TRANSFERRED (BIT #10)
1273          004000                  WRL=    4000            ;WRITE LOCK (BIT #11)
1274          010000                  MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
1275          020000                  PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
1276          040000                  ERR=    40000           ;COMPOSITE ERROR (BIT #14)
1277          100000                  ATA=    100000          ;ATTENTION ACTIVE (BIT #15)
1278
1279                                  ;ERROR REGISTER #01 (RPER1) (#02)
1280
1281          000001                  ILF=    1               ;ILLEGAL FUNCTION (BIT #0)
1282          000002                  ILR=    2               ;ILLEGAL REGISTER (BIT #1)
1283          000004                  RMR=    4               ;REGISTER MODIFICATION REFUSED (BIT #2)
1284          000010                  PAR=    10              ;PARITY ERROR (BIT #3)
1285          000020                  FER=    20              ;FORMAT ERROR (BIT #4)
1286          000040                  WCF=    40              ;WRITE CLOCK FAIL (BIT #5)
1287          000100                  ECH=    100             ;ECC HARD ERROR (BIT #6)
1288          000200                  HCE=    200             ;HEADER COMPARE ERROR (BIT #7)
1289          000400                  HCRC=   400             ;HEADER CRC ERROR (BIT #8)
1290          001000                  AOE=    1000            ;ADDRESS OVERFLOW ERROR (BIT #9)
1291          002000                  IAE=    2000            ;INVALID ADDRESS ERROR (BIT #10)
1292          004000                  WLE=    4000            ;WRITE LOCK ERROR (BIT #11)
1293          010000                  DTE=    10000           ;DRIVE TIMING ERROR (BIT #12)
1294          020000                  OPI=    20000           ;OPERATION INCOMPLETE (BIT #13)
1295          040000                  UNS=    40000           ;DRIVE UNSAFE (BIT #14)
1296          100000                  DCK=    100000          ;DATA CHECK ERROR (BIT 15)
1297
1298                                  ;MAINTAINABILITY REGISTER (RPMR)(#03)
1299
1300          000001                  DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
1301          000002                  MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
1302          000004                  MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
1303          000010                  MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1304          000020                  MRD=    20              ;MAINTAINABILITY READ (BIT #4)
1305          000040                  MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
1306          000200                  DTSY=   200             ;MAINTAINAB LITY SYNC DETECTED (BIT #7)
1307
1308                                  ;ATTENTION SUMMARY PSEUDO-REGISTER (RPAS) (#04)
1309
1310          000001                  AT0=    1               ;DEVICE 0 (BIT #0)
1311          000002                  AT1=    2               ;DEVICE 1 (BIT #1)
1312          000004                  AT2=    4               ;DEVICE 2 (BIT #2)
1313          000010                  AT3=    10              ;DEVICE 3 (BIT #3)
1314          000020                  AT4=    20              ;DEVICE 4 (BIT #4)
1315          000040                  AT5=    40              ;DEVICE 5 (BIT #5)
1316          000100                  AT6=    100             ;DEVICE 6 (BIT #6)
1317          000200                  AT7=    200             ;DEVICE 7 (BIT #7)
1318
1319                                  ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA) (#05)
1320                                  ;(EACH BIT IS CALLED BY BIT NUMBER)
```

```
1321
1322                                    ;DRIVE TYPE REGISTER (RPDT) (#06)
1323
1324    000001                          DT00=   1               ;DRIVE TYPE NUMBER BIT 1
1325    000002                          DT01=   2               ;DRIVE TYPE NUMBER BIT 2
1326    000004                          DT02=   4               ;DRIVE TYPE NUMBER BIT 3
1327    000010                          DT03=   10              ;DRIVE TYPE NUMBER BIT 4
1328    000020                          DT04=   20              ;DRIVE TYPE NUMBER BIT 5
1329    000040                          DT05=   40              ;DRIVE TYPE NUMBER BIT 6
1330    000100                          DT06=   100             ;DRIVE TYPE NUMBER BIT 7
1331    000200                          DT07=   200             ;DRIVE TYPE NUMBER BIT 8
1332    000400                          DT08=   400             ;DRIVE TYPE NUMBER BIT 9
1333    004000                          DRQ=    4000            ;DRIVE REQUEST REQUIRED (BIT #11)
1334    020000                          MOH=    20000           ;MOVING HEAD (BIT #13)
1335    040000                          TAP=    40000           ;TAPE DRIVE (BIT #14)
1336    100000                          NBA=    100000          ;NOT BLOCK ADDRESSED (BIT #15)
1337
1338                                    ;LOOK-AHEAD REGISTER (RPLA) (#07)
1339
1340    000001                          EXT1=   1               ;EXTENSION 1 (BIT #0)
1341    000002                          EXT2=   2               ;EXTENSION 2 (BIT #1)
1342    000004                          EXT4=   4               ;EXTENSION 3 (BIT #2)
1343    000010                          EXT10=  10              ;EXTENSION 4 (BIT #3)
1344    000020                          EXT20=  20              ;EXTENSION 5 (BIT #4)
1345    000040                          EXT40=  40              ;EXTENSION 6 (BIT #5)
1346    000100                          SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
1347    000200                          SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
1348    000400                          SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
1349    001000                          SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
1350    002000                          SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
1351    004000                          TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
1352    010000                          TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
1353    020000                          TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
1354    040000                          TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
1355    100000                          TRK20=  100000          ;TRACK FIELD 5 (BIT #15)
1356
1357                                    ;RP04 ERROR REGISTER #2 (RPER2) (#10)
1358
1359    000001                          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1360    000002                          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1361    000004                          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1362    000010                          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1363    000020                          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
1364    000040                          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1365    000100                          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1366    000200                          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
1367    000400                          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1368    001000                          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
1369    002000                          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1370    004000                          IXE=    4000            ;INDEX ERROR (BIT #11)
1371    010000                          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
1372    020000                          PLU=    20000           ;PLO UNSAFE (BIT #13)
1373    100000                          ACU=    100000          ;AC UNSAFE (BIT #15)
1374
1375                                    ;RP05/6 ERROR REGISTER #02 (RPER2) (#10)
1376
```

```
1377            000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1378            000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1379            000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1380            000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1381            000020          RAW=    20              ;READ AND WRITE (BIT #4)
1382            000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1383            000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1384            000200          ABS=    200             ;ABNORMAL STOP (BIT #7)
1385            000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1386            001000          MHS=    1000            ;MUTLTIPLE HEAD SELECT (BIT #9)
1387            002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1388            004000          IXE=    4000            ;INDEX ERROR (BIT #11)
1389            020000          PLU=    20000           ;PLO UNSAFE (BIT #12)
1390
1391                            ;OFFSET REGISTER (RPOF) (#11)
1392
1393            000001          OF25=   1               ;OFFSET 25 MICRO INCHES (BIT #0)
1394            000002          OF50=   2               ;OFFSET 50 MICRO INCHES (BIT #1)
1395            000004          OF100=  4               ;OFFSET 100 MICRO INCHES (BIT #2)
1396            000010          OF200=  10              ;OFFSET 200 MICRO INCHES (BIT #3)
1397            000020          OF400=  20              ;OFFSET 400 MICRO INCHES (BIT #4)
1398            000040          OF800=  40              ;OFFSET 800 MICRO INCHES (BIT #5)
1399            000200          OFREV=  200             ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1400            002000          HCI=    2000            ;HEADER COMPARE INHIBIT (BIT #10)
1401            004000          ECI=    4000            ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1402            010000          FMT22=  10000           ;FORMAT BIT (BIT #12
1403
1404                            ;DESIRED CYLINDER ADDRESS (RPCA) (#12)
1405                            ;(EACH BIT IS CALLED BY BIT NUMBER)
1406
1407                            ;CURRENT CYLINDER ADDRESS (RPCC) (#13)
1408                            ;(EACH BIT IS CALLED BY BIT NUMBER)
1409
1410                            ;SERIAL NUMBER REGISTER (RPSN) (#14)
1411                            ;(EACH IS CALLED BY BIT NUMBER)
1412
1413                            ;RP04 ERROR REGISTER #03 (RPER3) (#15)
1414
1415         000001             PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
1416         000002             VUF=    2               ;VELOCITY UNSAFE (BIT #1)
1417         000010             UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1418         000020             PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
1419         000040             ACL=    40              ;AC LOW (BIT #5)
1420         000100             DCL=    100             ;DC LOW (BIT #6)
1421         040000             SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
1422         100000             OCYL=   100000          ;OFF CYLINDER (BIT #15)
1423
1424                            ;RP05/6 ERROR REGISTER #03 (RPER3) (#15)
1425
1426         000001             DCU=    1               ;DC UNSAFE (BIT #0)
1427         000002             WAO=    2               ;WRITE AND OFFSET (BIT #1)
1428         000040             ACL=    40              ;AC LOW (BIT #5)
1429         000100             DCL=    100             ;DC LOW (BIT #6)
1430         020000             OPE=    20000           ;OPERATOR PLUG ERROR (BIT #13)
1431         040000             SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
1432         100000             OCYL=   100000          ;OFF CYLINDER ERROR (BIT #15)
```

```
                                ;ECC POSITION REGISTER (RPEC1) (#16)
                                ;(EACH BIT IS CALLED BY BIT NUMBER)

                                ;ECC PATTERN REGISTER (RPEC2) (#17)
                                ;(EACH BIT IS CALLED BY BIT NUMBER)

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

                                .SBTTL  DEFINITIONS OF THE RH11/RP04/5/6 ADDRESS INDEXES

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

 000000                         RPCS1=0                 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
 000002                         RPWC=2                  ;WORD COUNT REGISTER (NOT A DRIVE REG)
 000004                         RPBA=4                  ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
 000006                         RPDA=6                  ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
 000010                         RPCS2=10                ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
 000012                         RPDS1=12                ;DRIVE STATUS REGISTER (DRIVE REG 01)
 000014                         RPER1=14                ;ERROR REGISTER #1 (DRIVE REG. 02)
 000016                         RPAS=16                 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
 000020                         RPLA=20                 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
 000022                         RPDB=22                 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
 000024                         RPMR=24                 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
 000026                         RPDT=26                 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
 000030                         RPSN=30                 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
 000032                         RPOF=32                 ;OFFSET REGISTER (DRIVE REG. 11)
 000034                         RPCA=34                 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
 000036                         RPCC=36                 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
 000040                         RPER2=40                ;ERROR REGISTER #2 (DRIVE REG. 14)
 000042                         RPER3=42                ;ERROR REGISTER #3 (DRIVE REG. 15)
 000044                         RPEC1=44                ;ECC POSITION REGISTER (DRIVE REG. 16)
 000046                         RPEC2=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  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
 000200                                 $SVPC=.                 ;SAVE PC
 000046                                 .=46
 000046  056424                         $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
 000052                                 .=52
 000052  020000                         .WORD   20000           ;;2)SET LOC.52 TO 20000
 000200                                 .=$SVPC                 ;; RESTORE PC
```

```
1489                                  .SBTTL  STARTING ADDRESS = 200
1490
1491  000200  000137  002064                  JMP     START           ;START THE PROGRAM
1492
1493                                  .SBTTL  START THE PROGRAM AND CHANGE THE RH11 ADDRESS = 204
1494
1495  000204  000137  002072                  JMP     START1          ;START AND CHANGE THE RH11 ADDRESS
1496
1497
```

```
                                .SBTTL  COMMON TAGS

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

                    001100              .=1100
001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000      000000      $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
001102  000         000         $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103  000         000         $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000      000000      $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000      000000      $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000      000000      $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000      000000      $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114  000         000         $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115  001         001         $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000      000000      $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000      000000      $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000      000000      $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000      000000      $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000      000000      $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000      000000              .WORD   0       ;;RESERVED--NOT TO BE USED
001132  000000      000000              .WORD   0
001134  000         000         $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
001135  000         000         $INTAG: .BYTE   0       ;;INTERRUPT MODE INDICATOR
001136  000000      000000              .WORD   0
001140  177570      177570      SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
001142  177570      177570      DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
001144  177560      177560      $TKS:   177560          ;;TTY KBD STATUS
001146  177562      177562      $TKB:   177562          ;;TTY KBD BUFFER
001150  177564      177564      $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566      177566      $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000         000         $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002         002         $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012         012         $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000         000         $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000      000000      $REGAD: .WORD   0       ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REGO) WAS OBTAINED
001162  000000      000000      $REGO:  .WORD   0       ;;CONTAINS (($REGAD)+0)
001164  000000      000000      $TMP0:  .WORD   0       ;;USER DEFINED
001166  000000      000000      $TMP1:  .WORD   0       ;;USER DEFINED
001170  000000      000000      $TMP2:  .WORD   0       ;;USER DEFINED
001172  000000      000000      $TMP3:  .WORD   0       ;;USER DEFINED
001174  000000      000000      $TMP4:  .WORD   0       ;;USER DEFINED
001176  000000      000000      $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
001200  000000      000000      $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
001202  177607      177607      $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001206  077         077         $QUES:  .ASCII  /?/     ;;QUESTION MARK
001207  015         015         $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001210  000012      000012      $LF:    .ASCIZ  <12>    ;;LINE FEED
                                ;;**************************************************************
                    000015      CR      =       15
                    000012      LF      =       12
001212  172540      172540      $LKCSR: .WORD   172540  ;ADDR OF KW11-P STATUS REGISTER
001214  172542      172542      $LKCSB: .WORD   172542  ;ADDR OF KW11-P COUNTER BUFFER
```

```
1554  001216  000104                  $LPVEC: .WORD   104             ;ADDR OF KW11-P VECTOR
1555  001220  177546                  $LKS:   .WORD   177546          ;ADDR OF KW11-L STATUS REGISTER
1556  001222  000100                  $LLVEC: .WORD   100             ;ADDR OF KW11-L VECTOR
1557  001224  000000                  PORTA:  .WORD   0               ;ADDRESS OF PORT A
1558  001226  000000                  PORTB:  .WORD   0               ;ADDRESS OF PORT B
1559  001230  000000                  PORTC:  .WORD   0               ;ADDRESS OF DIFFERENT DRIVE
1560  001232  000000                  ASR1:   .WORD   0               ;ATA-A OR ATA-B = 1
1561  001234  000000                  PTNBR:  .WORD   0               ;CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
1562  001236  000000                  SEIZPT: .WORD   0               ;CONTAINS THE ADDRESS OF THE SEIZING PORT
1563  001240  000000                  OPPRT:  .WORD   0               ;CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
1564  001242  000000                  TSTNUM: .WORD   0               ;NUMBER OF THE CURRENT TEST
1565  001244  000000                  CKERR:  .WORD   0               ;IF -1, A REGISTER MISCOMPARISON OCCURRED
1566  001246  000000                  NOSEIZ: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
1567  001250  000000                  RELERR: .WORD   0               ;IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
1568  001252  000000                  TIME:   .WORD   0               ;ELAPSED TIME COUNTER
1569  001254  000000                  WATCH:  .WORD   0               ;WATCH DOG TIMER LOCATION
1570  001256  000000                  TIMEA:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
1571  001260  000000                  TIMEAP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
1572  001262  000000                  TIMEAM: .WORD   0               ;PORT A TIMEOUT VALUE - 25%
1573  001264  000000                  TIMEB:  .WORD   0               ;THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
1574  001266  000000                  TIMEBP: .WORD   0               ;PORT A TIMEOUT VALUE + 25%
1575  001270  000000                  TIMEBM: .WORD   0               ;PORT B TIME VALUE - 25%
1576  001272  000000                  TIMES:  .WORD   0               ;STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
1577  001274  000000                  KYBCTL: .WORD   0               ;SINGLE TEST INDICATOR
1578  001276  000000                  CHGADR: .WORD   0               ;CHANGE THE RH11 ADDRESS INDICATOR
1579
1580                                  ;;*****************************************************************
1581
1582                                  .SBTTL  RH11/RP04/5/6 UNIBUS AND VECTOR ADDRESSES
1583
1584                                  ;;*****************************************************************
1585
1586  001300  176700                  $RPADR: .WORD   176700          ;RH11/RP04/5/6 UNIBUS ADDRESS
1587  001302  000254                  $RPVEC: .WORD   254             ;RH11 INTERRUPT VECTOR ADDRESS
1588
```

```
                                .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


001304                          $ERRTB:

                                ;ERROR 1

001304  062506                          EM1                     ;WRONG DRIVE TYPE
001306  066544                          DH1
001310  070302                          DT1
001312  070552                          DF1

                                ;ERROR 2

001314  062527                          EM2                     ;DRIVE NOT ON LINE
001316  066544                          DH1
001320  070302                          DT1
001322  070552                          DF1

                                ;ERROR 3

001324  062551                          EM3                     ;SERIAL NUMBERS NOT THE SAME
001326  066615                          DH3
001330  070316                          DT3
001332  070552                          DF1

                                ;ERROR 4

001334  062633                          EM4                     ;DRIVE NOT SEIZED BY PORT 'N'
001336  066664                          DH4
001340  070364                          DT7
001342  070565                          DF7

                                ;ERROR 5

001344  062664                          EM5                     ;WRONG STATUS SEEN BY THE SEIZING PORT
001346  067007                          DH5
001350  070332                          DT5
001352  070557                          DF5

                                ;ERROR 6

001354  062732                          EM6                     ;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA
001356  067257                          DH13
001360  070404                          DT13
```

```
1645  001362  070557                        DF5
1646
1647                                  ;ERROR 7
1648
1649  001364  063032                        EM7             ;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1650  001366  067063                        DH7
1651  001370  070364                        DT7
1652  001372  070565                        DF7
1653
1654                                  ;ERROR 10
1655
1656  001374  063113                        EM10            ;REGISTER CONTENTS INCORRECT
1657  001376  067007                        DH5
1658  001400  070332                        DT5
1659  001402  070557                        DF5
1660
1661                                  ;ERROR 11
1662
1663  001404  063143                        EM11            ;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1664  001406  067206                        DH11
1665  001410  070302                        DT1
1666  001412  070552                        DF1
1667
1668                                  ;ERROR 12
1669
1670  001414  063227                        EM12            ;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1671  001416  067753                        DH36
1672  001420  070472                        DT37
1673  001422  070600                        DF36
1674
1675                                  ;ERROR 13
1676
1677  001424  063277                        EM13            ;'VOLUME VALID' BIT NOT SET BY READIN PRESET
1678  001426  067257                        DH13
1679  001430  070404                        DT13
1680  001432  070557                        DF5
1681
1682                                  ;ERROR 14
1683
1684  001434  063364                        EM14            ;'VOLUME VALID' SET ON THE OPPOSITE PORT
1685  001436  067257                        DH13
1686  001440  070404                        DT13
1687  001442  070557                        DF5
1688
1689                                  ;ERROR 15
1690
1691  001444  063427                        EM15            ;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1692  001446  067063                        DH7
1693  001450  070364                        DT7
1694  001452  070565                        DF7
1695
1696                                  ;ERROR 16
1697
1698  001454  063506                        EM16            ;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1699  001456  067063                        DH7
1700  001460  070364                        DT7
```

```
1701  001462  070565                        DF7
1702
1703                                  ;ERROR 17
1704
1705  001464  063561                        EM17          ;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
1706  001466  067063                        DH7
1707  001470  070364                        DT7
1708  001472  070565                        DF7
1709
1710                                  ;ERROR 20
1711
1712  001474  063640                        EM20          ;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
1713  001476  067753                        DH36
1714  001500  070472                        DT37
1715  001502  070600                        DF36
1716
1717                                  ;ERROR 21
1718
1719  001504  063720                        EM21          ;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
1720  001506  067753                        DH36
1721  001510  070472                        DT37
1722  001512  070600                        DF36
1723
1724                                  ;ERROR 22
1725
1726  001514  063773                        EM22          ;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
1727  001516  067377                        DH22
1728  001520  070422                        DT22
1729  001522  070574                        DF31
1730
1731                                  ;ERROR 23
1732
1733  001524  064060                        EM23          ;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
1734  001526  067475                        DH23
1735  001530  070434                        DT23
1736  001532  070552                        DF1
1737
1738                                  ;ERROR 24
1739
1740  001534  064126                        EM24          ;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1741  001536  067475                        DH23
1742  001540  070434                        DT23
1743  001542  070552                        DF1
1744
1745
1746                                  ;ERROR 25
1747
1748  001544  064205                        EM25          ;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1749  001546  067753                        DH36
1750  001550  070462                        DT36
1751  001552  070600                        DF36
1752
1753
1754                                  ;ERROR 26
1755
1756  001554  064250                        EM26          ;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
```

```
1757  001556  067377                          DH22
1758  001560  070422                          DT22
1759  001562  070574                          DF31
1760
1761                                  ;ERROR 27
1762
1763  001564  064335                          EM27            ;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1764  001566  067063                          DH7
1765  001570  070364                          DT7
1766  001572  070565                          DF7
1767
1768                                  ;ERROR 30
1769
1770  001574  064413                          EM30            ;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
1771  001576  067753                          DH36
1772  001600  070462                          DT36
1773  001602  070600                          DF36
1774
1775                                  ;ERROR 31
1776
1777  001604  064510                          EM31            ;DRIVE NOT SEIZED BY PORT AFTER RELEASE WIITH REQUEST SE
1778  001606  067654                          DH31
1779  001610  070450                          DT31
1780  001612  070574                          DF31
1781
1782                                  ;ERROR 32
1783
1784  001614  064565                          EM32            ;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1785  001616  067007                          DH5
1786  001620  070332                          DT5
1787  001622  070557                          DF5
1788
1789                                  ;ERROR 33
1790
1791  001624  064636                          EM33            ;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1792  001626  067753                          DH36
1793  001630  070462                          DT36
1794  001632  070600                          DF36
1795
1796                                  ;ERROR 34
1797
1798  001634  064740                          EM34            ;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1799  001636  067753                          DH36
1800  001640  070462                          DT36
1801  001642  070600                          DF36
1802
1803                                  ;ERROR 35
1804
1805  001644  065043                          EM35            ;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
1806  001646  067753                          DH36
1807  001650  070472                          DT37
1808  001652  070600                          DF36
1809
1810                                  ;ERROR 36
1811
1812  001654  065122                          EM36            ;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
```

ERROR POINTER TABLE

```
1813  001656  067753                  DH36
1814  001660  070462                  DT36
1815  001662  070600                  DF36
1816
1817                            ;ERROR 37
1818
1819  001664  065174                  EM37            ;DRIVE IS NON-EXISTENT
1820  001666  067753                  DH36
1821  001670  070472                  DT37
1822  001672  070600                  DF36
1823
1824                            ;ERROR 40
1825
1826  001674  065242                  EM40            ;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1827  001676  066544                  DH1
1828  001700  070434                  DT23
1829  001702  070552                  DF1
1830
1831                            ;ERROR 41
1832
1833  001704  065317                  EM41            ;TIMEOUT CLEARED ATTENTION BIT
1834  001706  067475                  DH23
1835  001710  070434                  DT23
1836  001712  070552                  DF1
1837
1838                            ;ERROR 42
1839
1840  001714  065361                  EM42            ;DRIVE NOT IN NEUTRAL OR SEIZED
1841  001716  070002                  DH42
1842  001720  070502                  DT42
1843  001722  070603                  DF42
1844
1845                            ;ERROR 43
1846
1847  001724  065447                  EM43            ;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1848  001726  070002                  DH42
1849  001730  070502                  DT42
1850  001732  070603                  DF42
1851
1852                            ;ERROR 44
1853
1854  001734  065524                  EM44            ;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1855  001736  070021                  DH44
1856  001740  070450                  DT31
1857  001742  070574                  DF31
1858
1859                            ;ERROR 45
1860
1861  001744  065601                  EM45            ;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1862  001746  066544                  DH1
1863  001750  070302                  DT1
1864  001752  070552                  DF1
1865
1866                            ;ERROR 46
1867
1868  001754  065660                  EM46            ;CAN'T ACCESS DRIVE THROUGH EITHER PORT
```

ERROR POINTER TABLE

```
1869  001756  070117                  DH46
1870  001760  070510                  DT46
1871  001762  070574                  DF31
1872
1873                            ;ERROR 47
1874
1875  001764  065727                  EM47          ;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
1876  001766  067475                  DH23
1877  001770  070434                  DT23
1878  001772  070552                  DF1
1879
1880                            ;ERROR 50
1881
1882  001774  066015                  EM50          ;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
1883  001776  067257                  DH13
1884  002000  070404                  DT13
1885  002002  070557                  DF5
1886
1887                            ;ERROR 51
1888
1889  002004  066100                  EM51          ;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1890  002006  067007                  DH5
1891  002010  070332                  DT5
1892  002012  070557                  DF5
1893
1894                            ;ERROR 52
1895
1896  002014  066163                  EM52          ;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1897  002016  067257                  DH13
1898  002020  070404                  DT13
1899  002022  070557                  DF5
1900
1901                            ;ERROR 53          ;CAN'T READ ATTN BIT FROM OPPOSITE PORT
1902
1903  002024  066261                  EM53
1904  002026  067475                  DH23
1905  002030  070302                  DT1
1906  002032  070552                  DF1
1907
1908                            ;ERROR 54
1909
1910  002034  066342                  EM54          ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1911  002036  067377                  DH22
1912  002040  070522                  DT54
1913  002042  070574                  DF31
1914
1915                            ;ERROR 55
1916
1917  002044  066435                  EM55          ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1918  002046  070215                  DH55
1919  002050  070534                  DT55
1920  002052  070605                  DF55
1921
1922                            ;ERROR 56
1923
1924  002054  066502                  EM56          ;RH11 DIDN'T RESPOND TO ADDRESSING
```

```
 1925  002056  070273                          DH56
 1926  002060  070546                          DT56
 1927  002062  070611                          DF56
 1928
 1929
 1930
 1931                                  ;;*****************************************************************
 1932
 1933                                  .SBTTL  STARTUP AND INITIALIZATION ROUTINES
 1934
 1935                                  ;;*****************************************************************
 1936
 1937  002064  005037  001276          START:  CLR     CHGADR          ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
 1938  002070  000403                          BR      START2          ;GO TO THE START
 1939  002072  012737  177777  001276  START1: MOV     #-1,CHGADR      ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
 1940  002100  000005                  START2: RESET                   ;CLEAR THE BUS
 1941                                  .SBTTL  INITIALIZE THE COMMON TAGS
 1942                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
 1943  002102  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
 1944  002106  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
 1945  002110  022706  001140                  CMP     #$WR,R6 ;;DONE?
 1946  002114  001374                          BNE     .-6             ;;LOOP BACK IF NO
 1947  002116  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
 1948                                  ;;INITIALIZE A FEW VECTORS
 1949  002122  012737  056670  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
 1950  002130  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
 1951  002136  012737  057122  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
 1952  002144  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
 1953  002152  012737  061722  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
 1954  002160  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
 1955  002166  013737  056270  056262          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
 1956  002174  005037  001176                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
 1957  002200  005037  001200                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
 1958  002204  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
 1959  002212  012737  002212  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
 1960  002220  012737  002220  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
 1961                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
 1962                                  ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
 1963  002226  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
 1964  002232  012737  002266  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
 1965  002240  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
 1966  002246  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
 1967  002254  022777  177777  176656          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
 1968  002262  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
 1969                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
 1970  002264  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
 1971  002266  012716  002274          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
 1972  002272  000002                          RTI
 1973  002274  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
 1974  002302  012737  000174  001142          MOV     #DISPREG,DISPLAY
 1975  002310  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
 1976
 1977  002314  005227  177777                  INC     #-1             ;FIRST START ?
 1978  002320  001002                          BNE     1$              ;BR IF NOT
 1979  002322  104401  062010                  TYPE    ,TITLE          ;TYPE PROGRAM NAME
 1980  002326  004737  060332          1$:     JSR     PC,$TKINT       ;SETUP THE TTY KEYBOARD
```

```
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
002332  005737  000042                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
002336  001006                                BNE     67$             ;;BRANCH IF YES
002340  023727  001140  000176                CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
002346  001005                                BNE     68$             ;;BRANCH IF NO
002350  104406                                GTSWR                   ;;GET SOFT-SWR SETTINGS
002352  000403                                BR      68$
002354  112737  000001  001134  67$:          MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
002362                                68$:
002362  004737  002754                        JSR     PC,CHANGE       ;CHECK/CHANGE THE RH11 ADDRESS
002366  104401  062117                        TYPE    ,ENTERA         ;ENTER DRIVE ADDRESS
002372  104412                                RDOCT                   ;GET THE ADDRESS
002374  012637  001224                        MOV     (SP)+,PORTA     ;STORE THE ADDRESS
002400  023727  001224  000007                CMP     PORTA,#7        ;SEE IF ADDRESS TOO LARGE
002406  101403                                BLOS    2$              ;BR IF NOT
002410  104401  062147                        TYPE    ,ADRERR         ;TYPE ADDRESS ERROR MESSAGE
002414  000744                                BR      1$              ;TRY AGAIN
002416  013737  001224  001226  2$:           MOV     PORTA,PORTB     ;GENERATE THE PORT B ADDRESS
002424  005237  001226                        INC     PORTB           ;INCREMENT THE ADDRESS
002430  042737  000016  001226                BIC     #16,PORTB       ;LEAVE BIT 0
002436  013746  001224                        MOV     PORTA,-(SP)     ;PUT PORT A ADDRESS ON THE STACK
002442  042716  177771                        BIC     #^C6,(SP)       ;SAVE BITS 1 & 2
002446  052637  001226                        BIS     (SP)+,PORTB     ;SET BITS 1 & 2 IN PORT B ADDRESS
002452  104401  062171                        TYPE    ,PORTAIS        ;'PORT A ADDRESS IS '
002456  013746  001224                        MOV     PORTA,-(SP)     ;;SAVE PORTA FOR TYPEOUT
                                                                        ;;TYPE PORT A ADDRESS
002462  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
002464     001                                .BYTE   1               ;;TYPE 1 DIGIT(S)
002465     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
002466  104401  062217                        TYPE    ,PORTBIS        ;'PORT B ADDRESS IS '
002472  013746  001226                        MOV     PORTB,-(SP)     ;;SAVE PORTB FOR TYPEOUT
                                                                        ;;TYPE PORT B ADDRESS
002476  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
002500     001                                .BYTE   1               ;;TYPE 1 DIGIT(S)
002501     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
002502  104401  001207                        TYPE    ,$CRLF          ;ANOTHER CR-LF
002506  013737  001224  001230                MOV     PORTA,PORTC     ;GENERATE ADDRESS OF DRIVE NOT TESTED
002514  062737  000006  001230                ADD     #6,PORTC        ;COMPLEMENT SOME BITS
002522  042737  177770  001230                BIC     #^C7,PORTC      ;SAVE ONLY LOWER BITS
002530  013701  001224                        MOV     PORTA,R1        ;USE PORT A ADDRESS AS INDEX
002534  116137  070726  001232                MOVB    ATABIT(R1),ASR1  ;GET ATTENTION BIT FOR DRIVE
002542  005037  001256                        CLR     TIMEA           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
002546  005037  001260                        CLR     TIMEAP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
002552  005037  001264                        CLR     TIMEB           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
002556  005037  001266                        CLR     TIMEBP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
002562  004737  056444                        JSR     PC,CKCLK        ;SETUP CLOCK
002566  000137  002602                        JMP     EXEC            ;CLOCK HAS BEEN STARTED
002572  104401  062245                        TYPE    ,NOCLOCK        ;NO CLOCK ON SYSTEM
002576  000000                        3$:     HALT                    ;FATAL ERROR
002600  000776                                BR      3$              ;INTERLOCK THE HALT

                                        ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR

002602  000005                        EXEC:   RESET                   ;CLEAR EVERYTHING
002604  005037  177776                        CLR     PS              ;CLEAR THE PROCESSOR STATUS WORD
002610  104401  001207                        TYPE    ,$CRLF          ;CR-LF
```

```
                                        GET VALUE FOR SOFTWARE SWITCH REGISTER

 2037  002614  013700  001300                    MOV     $RPADR,R0       ;RH11 ADDRESS FOR INDEXING
 2038  002620  012706  001100                    MOV     #STACK,SP       ;LOAD STACK POINTER
 2039  002624  004737  056444                    JSR     PC,CKCLK        ;START THE CLOCK
 2040  002630  000240                            NOP                     ;RETURN IF NO CLOCK
 2041  002632  004737  060332                    JSR     PC,$TKINT       ;INITIALIZE THE KEYBOARD
 2042  002636  005037  001274                    CLR     KYBCTL          ;CLEAR SINGLE TEST INDICATOR
 2043  002642  005037  001100                    CLR     $PASS           ;CLEAR THE PASS COUNT
 2044  002646  112737  000001  001115            MOVB    #1,$ERMAX       ;SET ERROR MAX TO 1
 2045  002654  012737  002654  001106            MOV     #.,$LPADR       ;INITIAL SETTING FOR LOOP ADDRESS
 2046  002662  012737  002662  001110            MOV     #.,$LPERR       ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
 2047  002670  104401  062314          1$:       TYPE    ,TESTNO         ;ASK FOR TEST NUMBER
 2048  002674  104412                            RDOCT                   ;GET THE NUMBER
 2049  002676  012601                            MOV     (SP)+,R1        ;PUT ENTRY INTO R1
 2050  002700  001002                            BNE     2$              ;BR IF NOT ZERO
 2051  002702  000137  003070                    JMP     TST1            ;ENTER ZERO - PERFORM ALL TESTS
 2052  002706  020137  070736          2$:       CMP     R1,MAXTN        ;SEE IF NUMBER GREATER THAN MAXIMUM
 2053  002712  003403                            BLE     3$              ;BR IF LESS OR EQUAL
 2054  002714  104401  062334                    TYPE    ,BADNO          ;BAD ENTRY
 2055  002720  000763                            BR      1$              ;TRY AGAIN
 2056  002722  005301          3$:       DEC     R1                      ;DECREMENT ENTRY
 2057  002724  006301                            ASL     R1              ;SHIFT IT LEFT
 2058  002726  016137  070612  002752            MOV     TSTADR(R1),4$   ;GET THE TEST ADDRESS
 2059  002734  005237  001274                    INC     KYBCTL          ;SET SINGLE TEST INDICATOR
 2060  002740  012737  000001  001104            MOV     #1,$ICNT        ;PRESET ITERATION COUNT
 2061  002746  000177  000000                    JMP     @4$             ;GO TO THE SELECTED TEST
 2062  002752  000000          4$:       .WORD   0                       ;TEST ADDRESS GOES HERE
 2063
 2064                                    ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
 2065
 2066  002754  005737  001276          CHANGE: TST     CHGADR          ;CHANGE THE ADDRESS ?
 2067  002760  001421                            BEQ     3$              ;BR IF NOT
 2068  002762  005037  001276                    CLR     CHGADR          ;CLEAR THE INDICATOR
 2069  002766  104401  062374          1$:       TYPE    ,ADDRIS         ;TYPE OUT WHAT THE PRESENT ADDRESS IS
 2070  002772  013746  001300                    MOV     $RPADR,-(SP)    ;PUT THE ADDRESS ON THE STACK
 2071  002776  104402                            TYPOC                   ;TYPE THE ACTUAL ADDRESS
 2072  003000  104401  001207                    TYPE    ,$CRLF          ;CR-LF
 2073  003004  104401  062454                    TYPE    ,NTRH11         ;ASK FOR NEW ADDRESS
 2074  003010  104412                            RDOCT
 2075  003012  005716                            TST     (SP)            ;0 OR 'CR' ENTERED ?
 2076  003014  001402                            BEQ     2$              ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
 2077  003016  011637  001300                    MOV     (SP),$RPADR     ;NEW RH11 ADDRESS
 2078  003022  005726          2$:       TST     (SP)+                   ;CORRECT THE STACK POINTER
 2079  003024  012737  003044  000004    3$:     MOV     #4$,@#4         ;LOAD TRAP ADDRESS
 2080  003032  013700  001300                    MOV     $RPADR,R0       ;RH11 ADDRESS
 2081  003036  005760  000002                    TST     RPWC(R0)        ;SEE IF RH11 RESPONDS AT THAT ADDRESS
 2082  003042  000404                            BR      5$              ;BR, RH11 ALIVE AT PRESENT ADDRESS
 2083  003044  104056          4$:       ERROR   56                      ;NO RESPONSE TO ADDRESS
 2084  003046  062706  000004                    ADD     #4,SP           ;RESET THE STACK POINTER
 2085  003052  000745                            BR      1$              ;GET ADDRESS AGAIN
 2086  003054  012737  000006  000004    5$:     MOV     #6,@#4          ;RESTORE THE VECTOR
 2087  003062  000207                            RTS     PC              ;RETURN
 2088
 2089                                    ;;*******************************************************************
 2090
 2091                                    .SBTTL  *** TESTS ***
 2092
```

```
                                        *** TESTS ***

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

                                        TST1AA: MOV     $RPADR,R0       ;;RESTORE R0 AFTER END OF PASS

                                        ;;*****************************************************************
                                        ;*TEST 1         DRIVE ACCESS TEST
                                        ;*
                                        ;*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
                                        ;*
                                        ;*  A.  SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
                                        ;*      DRIVE IS A DUAL PORT RP04, THAT THE DRIVE IS ONLINE (RPDS1 HAS
                                        ;*      'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL
                                        ;*      NUMBER READ THROUGH BOTH PORTS IS THE SAME.
                                        ;*
                                        ;*  B.  THE TEST IS REPEATED THROUGH BOTH PORTS.
                                        ;*
                                        ;;*****************************************************************
003070                                  TST1:
003070  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
003074  001406                                  BEQ     2$              ;BR IF NOT
003076  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
003100  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
003104  012737  177777  001274    1$:           MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
003112  112737  000001  001102    2$:           MOVB    #1,$TSTNM       ;TEST NUMBER
003120  012737  003142  001106                  MOV     #TEST1,$LPADR   ;LOAD LOOP ON TEST ADDRESS
003126  012737  003142  001110                  MOV     #TEST1,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
003134  012737  000001  001176                  MOV     #1,$TIMES       ;;DO 1 ITERATION
003142  012706  001100                TEST1:    MOV     #STACK,SP       ;LOAD THE STACK POINTER

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

003146  113760  001224  000010                  MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
003154  013737  001224  001234                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003162  005760  000012                          TST     RPDS1(R0)       ;SEE IF DRIVE (PORT A) PRESENT
003166  005037  001244                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
003172  016037  000010  001126                  MOV     RPCS2(R0),$BDDAT  ;GET CONTENTS OF RPCS2
003200  012737  000010  001122                  MOV     #RPCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003206  060037  001122                          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
003212  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
003216  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
003224  042737  167777  001164                  BIC     #^CNED,$TMP0  ;SAVE SPECIFIED BITS
003232  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
003240  001414                                  BEQ     64$             ;BR IF OK
003242  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
003250  042737  010000  001174                  BIC     #NED,$TMP4      ;CLEAR THE MASKED BITS
003256  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
003264  104037                                  ERROR   37              ;TYPE MESSAGE 37
003266  005137  001244                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
003272  000240                        64$:      NOP
003274  005737  001244                          TST     CKERR           ;WAS 'NED' SET ?
003300  001403                                  BEQ     .+10            ;BR IF NOT
003302  012760  000040  000010                  MOV     #CLR,RPCS2(R0)  ;ISSUE MASSBUS INIT TO CLEAR 'NED'
003310  113760  001226  000010                  MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
003316  013737  001226  001234                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
2149  003324  005760  000012                  TST     RPDS1(R0)          ;SEE IF DRIVE (PORT B) PRESENT
2150  003330  005037  001244                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
2151  003334  016037  000010  001126          MOV     RPCS2(R0),$BDDAT   ;GET CONTENTS OF RPCS2
2152  003342  012737  000010  001122          MOV     #RPCS2,$BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2153  003350  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
2154  003354  005037  001124                  CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
2155  003360  013737  001126  001164          MOV     $BDDAT,$TMP0       ;MOVE REGISTER CONTENTS TO '$TMP0'
2156  003366  042737  167777  001164          BIC     #^CNED,$TMP0       ;SAVE SPECIFIED BITS
2157  003374  023737  001124  001164          CMP     $GDDAT,$TMP0       ;COMPARE THE BITS
2158  003402  001414                          BEQ     66$                ;BR IF OK
2159  003404  013737  001126  001174          MOV     $BDDAT,$TMP4       ;COPY 'BAD DATA'
2160  003412  042737  010000  001174          BIC     #NED,$TMP4         ;CLEAR THE MASKED BITS
2161  003420  053737  001174  001124          BIS     $TMP4,$GDDAT       ;'OR' WITH GOOD DATA FOR TYPEOUT
2162  003426  104037                          ERROR   37                 ;TYPE MESSAGE 37
2163  003430  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
2164  003434  000240                  66$:    NOP
2165  003436  005737  001244                  TST     CKERR              ;WAS 'NED' SET ?
2166  003442  001403                          BEQ     .+10               ;BR IF NOT
2167  003444  012760  000040  000010          MOV     #CLR,RPCS2(R0)     ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2168
2169                                  ;;*****************************************************************
2170                                  ;CONFIRM THAT DRIVE IS AN RP04/5/6 AND IS DUAL PORT
2171
2172  003452  113760  001224  000010          MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A
2173  003460  013737  001224  001234          MOV     PORTA,PTNBR        ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2174  003466  005037  001244                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
2175  003472  016037  000026  001126          MOV     RPDT(R0),$BDDAT    ;GET CONTENTS OF RPDT
2176  003500  012737  000026  001122          MOV     #RPDT,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2177  003506  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
2178  003512  012737  024020  001124          MOV     #24020,$GDDAT      ;WHAT REGISTER SHOULD BE
2179  003520  013737  001126  001164          MOV     $BDDAT,$TMP0       ;MOVE REGISTER CONTENTS TO '$TMP0'
2180  003526  042737  000003  001164          BIC     #^C177774,$TMP0    ;SAVE SPECIFIED BITS
2181  003534  023737  001124  001164          CMP     $GDDAT,$TMP0       ;COMPARE THE BITS
2182  003542  001414                          BEQ     68$                ;BR IF OK
2183  003544  013737  001126  001174          MOV     $BDDAT,$TMP4       ;COPY 'BAD DATA'
2184  003552  042737  177774  001174          BIC     #177774,$TMP4      ;CLEAR THE MASKED BITS
2185  003560  053737  001174  001124          BIS     $TMP4,$GDDAT       ;'OR' WITH GOOD DATA FOR TYPEOUT
2186  003566  104001                          ERROR   1                  ;TYPE MESSAGE 1
2187  003570  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
2188  003574  000240                  68$:    NOP
2189  003576  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B
2190  003604  013737  001226  001234          MOV     PORTB,PTNBR        ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2191  003612  005037  001244                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
2192  003616  016037  000026  001126          MOV     RPDT(R0),$BDDAT    ;GET CONTENTS OF RPDT
2193  003624  012737  000026  001122          MOV     #RPDT,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2194  003632  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
2195  003636  012737  024020  001124          MOV     #24020,$GDDAT      ;WHAT REGISTER SHOULD BE
2196  003644  013737  001126  001164          MOV     $BDDAT,$TMP0       ;MOVE REGISTER CONTENTS TO '$TMP0'
2197  003652  042737  000003  001164          BIC     #^C177774,$TMP0    ;SAVE SPECIFIED BITS
2198  003660  023737  001124  001164          CMP     $GDDAT,$TMP0       ;COMPARE THE BITS
2199  003666  001414                          BEQ     70$                ;BR IF OK
2200  003670  013737  001126  001174          MOV     $BDDAT,$TMP4       ;COPY 'BAD DATA'
2201  003676  042737  177774  001174          BIC     #177774,$TMP4      ;CLEAR THE MASKED BITS
2202  003704  053737  001174  001124          BIS     $TMP4,$GDDAT       ;'OR' WITH GOOD DATA FOR TYPEOUT
2203  003712  104001                          ERROR   1                  ;TYPE MESSAGE 1
2204  003714  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
2205  003720  000240                    70$:    NOP
2206
2207                                    ;;*****************************************************************
2208                                    ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
2209
2210  003722  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
2211  003730  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2212  003736  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2213  003742  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2214  003750  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2215  003756  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2216  003762  012737  001000  001124          MOV     #PGM,$GDDAT  ;WHAT REGISTER SHOULD BE
2217  003770  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2218  003776  042737  176777  001164          BIC     #^CPGM,$TMP0  ;SAVE SPECIFIED BITS
2219  004004  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
2220  004012  001414                          BEQ     72$              ;BR IF OK
2221  004014  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
2222  004022  042737  001000  001174          BIC     #PGM,$TMP4       ;CLEAR THE MASKED BITS
2223  004030  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
2224  004036  104045                          ERROR   45               ;TYPE MESSAGE 45
2225  004040  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2226  004044  000240                    72$:    NOP
2227  004046  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2228  004052  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2229  004060  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2230  004066  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2231  004072  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2232  004100  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2233  004106  042737  167177  001164          BIC     #^C10600,$TMP0   ;SAVE SPECIFIED BITS
2234  004114  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
2235  004122  001414                          BEQ     74$              ;BR IF OK
2236  004124  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
2237  004132  042737  010600  001174          BIC     #10600,$TMP4     ;CLEAR THE MASKED BITS
2238  004140  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
2239  004146  104002                          ERROR   2                ;TYPE MESSAGE 2
2240  004150  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2241  004154  000240                    74$:    NOP
2242  004156  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
2243  004164  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2244  004172  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2245  004176  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2246  004204  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2247  004212  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2248  004216  012737  001000  001124          MOV     #PGM,$GDDAT  ;WHAT REGISTER SHOULD BE
2249  004224  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
2250  004232  042737  176777  001164          BIC     #^CPGM,$TMP0  ;SAVE SPECIFIED BITS
2251  004240  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
2252  004246  001414                          BEQ     76$              ;BR IF OK
2253  004250  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
2254  004256  042737  001000  001174          BIC     #PGM,$TMP4       ;CLEAR THE MASKED BITS
2255  004264  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
2256  004272  104045                          ERROR   45               ;TYPE MESSAGE 45
2257  004274  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
2258  004300  000240                    76$:    NOP
2259  004302  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2260  004306  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
```

```
2261  004314  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2262  004322  060037  001122                  ADD     R0,$BDADR      ;ADD RH11 BASE ADDRESS
2263  004326  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2264  004334  013737  001126  001164          MOV     $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
2265  004342  042737  167177  001164          BIC     #^C10600,$TMP0 ;SAVE SPECIFIED BITS
2266  004350  023737  001124  001164          CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
2267  004356  001414                          BEQ     78$            ;BR IF OK
2268  004360  013737  001126  001174          MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
2269  004366  042737  010600  001174          BIC     #10600,$TMP4   ;CLEAR THE MASKED BITS
2270  004374  053737  001174  001124          BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
2271  004402  104002                          ERROR   2              ;TYPE MESSAGE 2
2272  004404  005137  001244                  COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
2273  004410  000240                  78$:    NOP
2274
2275                                  ;;*****************************************************************
2276                                  ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
2277
2278  004412  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2279  004420  016037  000030  001124          MOV     RPSN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
2280  004426  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
2281  004434  016037  000030  001126          MOV     RPSN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
2282  004442  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ARE THEY THE SAME ?
2283  004450  001406                          BEQ     1$              ;BR IF THEY ARE
2284  004452  104003                          ERROR   3               ;REPORT THE ERROR
2285  004454  032777  100000  174456          BIT     #SW15,@SWR      ;HALT ON ERROR ?
2286  004462  001001                          BNE     1$              ;BR IF SET - PROGRAM HAS ALREADY HALTED
2287  004464  000000                          HALT                    ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
2288  004466  000004                  1$:     SCOPE                   ;LOOP ?
2289
2290
2291                                  ;;*****************************************************************
2292                                  ;*TEST 2        PORT 'A' SEIZE/TIMEOUT TEST
2293                                  ;*
2294                                  ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
2295                                  ;*      THE DRIVE.
2296                                  ;*
2297                                  ;*  A.  WRITE 0'S INTO RPDS1 THROUGH PORT 'A'; VERIFY THAT THE DRIVE
2298                                  ;*      HAS BEEN SEIZED.
2299                                  ;*
2300                                  ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'B';
2301                                  ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
2302                                  ;*
2303                                  ;*  C.  WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
2304                                  ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
2305                                  ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
2306                                  ;*      NEUTRAL.
2307                                  ;*
2308                                  ;*****************************************************************
2309  004470                  TST2:
2310  004470  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2311  004474  001406                          BEQ     2$              ;BR IF NOT
2312  004476  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
2313  004500  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
2314  004504  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
2315  004512  112737  000002  001102  2$:     MOVB    #2,$TSTNM       ;TEST NUMBER
2316  004520  012737  004542  001106          MOV     #TEST2,$LPADR   ;LOAD LOOP ON TEST ADDRESS
```

```
2317  004526  012737  004542  001110          MOV     #TEST2,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
2318  004534  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
2319  004542  012706  001100          TEST2:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2320  004546  012737  000240  177776          MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
2321  004554  005037  001256                  CLR     TIMEA           ;CLEAR TIMEOUT VALUE FOR PORT A
2322  004560  005037  001260                  CLR     TIMEAP  ;CLEAR UPPER TIMEOUT TOLERANCE
2323  004564  005037  001262                  CLR     TIMEAM  ;CLEAR LOWER TIMEOUT TOLERANCE
2324
2325                                  ;;**********************************************************************
2326                                  ;START THE TIMER
2327
2328  004570  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
2329  004574  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
2330
2331                                  ;;**********************************************************************
2332
2333                                          ;SEIZE THE DRIVE THROUGH PORT A
2334
2335  004602  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2336  004610  013737  001224  001236          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
2337  004616  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
2338  004622  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
2339  004630  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2340  004636  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
2341  004644  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
2342  004652  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
2343  004656  062737  000012  001122          ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
2344  004664  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
2345  004670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
2346  004676  001403                          BEQ     64$             ;BR IF IT IS
2347  004700  104004                          ERROR   4               ;REPORT THE ERROR
2348  004702  000137  006066                  JMP     5$              ;BYPASS REST OF THE SUBTEST
2349  004706                          64$:
2350  004706  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2351  004714  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2352  004722  016037  000012  001126          MOV     RPDS1(R0) $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
2353  004730  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
2354  004736  013737  001124  001166          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
2355  004744  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
2356  004750  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
2357  004756  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
2358  004764  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
2359  004772  001401                          BEQ     65$             ;BR IF THEY ARE
2360  004774  104005                          ERROR 5                 ;REPORT THE ERROR
2361  004776  000240                  65$:    NOP
2362
2363                                  ;;**********************************************************************
2364                                  ;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK
2365
2366  005000  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
2367  005006  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2368  005014  016046  000046                  MOV     RPEC2(R0),-(SP) ;STORE REGISTER RPEC2, PORT B, FOR CHECK
2369  005020  016046  000044                  MOV     RPEC1(R0),-(SP) ;STORE REGISTER RPEC1, PORT B, FOR CHECK
2370  005024  016046  000042                  MOV     RPER3(R0),-(SP) ;STORE REGISTER RPER3, PORT B, FOR CHECK
2371  005030  016046  000030                  MOV     RPSN(R0),-(SP)  ;STORE REGISTER RPSN, PORT B, FOR CHECK
2372  005034  016046  000036                  MOV     RPCC(R0),-(SP)  ;STORE REGISTER RPCC, PORT B, FOR CHECK
```

```
2373  005040  016046  000034                MOV     RPCA(R0),-(SP)  ;STORE REGISTER RPCA, PORT B, FOR CHECK
2374  005044  016046  000032                MOV     RPOF(R0),-(SP)  ;STORE REGISTER RPOF, PORT B, FOR CHECK
2375  005050  016046  000040                MOV     RPER2(R0),-(SP) ;STORE REGISTER RPER2, PORT B, FOR CHECK
2376  005054  016046  000020                MOV     RPLA(R0),-(SP)  ;STORE REGISTER RPLA, PORT B, FOR CHECK
2377  005060  016046  000026                MOV     RPDT(R0),-(SP)  ;STORE REGISTER RPDT, PORT B, FOR CHECK
2378  005064  016046  000006                MOV     RPDA(R0),-(SP)  ;STORE REGISTER RPDA, PORT B, FOR CHECK
2379  005070  016046  000024                MOV     RPMR(R0),-(SP)  ;STORE REGISTER RPMR, PORT B, FOR CHECK
2380  005074  016046  000014                MOV     RPER1(R0),-(SP) ;STORE REGISTER RPER1, PORT B, FOR CHECK
2381
2382                                  ;;*****************************************************************
2383                                  ;WAIT FOR PORT A TO TIMEOUT
2384
2385  005100  005760  000012        1$:     TST     RPDS1(R0)       ;WAIT FOR THE DRIVE TO TIMEOUT
2386  005104  001006                        BNE     2$              ;BR WHEN TIMEOUT OCCURS
2387  005106  005737  001254                TST     WATCH           ;CHECK WATCH
2388  005112  001372                        BNE     1$              ;BR IF NOT ZERO
2389  005114  104036                        ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
2390  005116  000137  005506                JMP     4$              ;BYPASS TIMEOUT TIME CHECK
2391  005122  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS   ;SET PRIORITY TO 7 TO STOP CLOCK
2392  005130  013737  001252  001256        MOV     TIME,TIMEA      ;SAVE THE ELAPSED TIME FOR PORT A
2393  005136  004537  056630                JSR     R5,TOLER        ;CALCULATE THE TOLERANCE
2394  005142  001256                        .WORD   TIMEA           ;TIMEOUT VALUE FOR PORT A
2395  005144  012637  001260                MOV     (SP)+,TIMEAP    ;+25% TOLERANCE
2396  005150  012637  001262                MOV     (SP)+,TIMEAM    ;-25% TOLERANCE
2397
2398                                  ;;*****************************************************************
2399                                  ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
2400
2401  005154  023727  001252  000764        CMP     TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
2402  005162  103001                        BHIS    3$              ;BR IF IT  WAS
2403  005164  104055                        ERROR   55              ;REPORT TIMEOUT TOO SHORT
2404
2405                                  ;;*****************************************************************
2406                                  ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
2407
2408  005166  012737  000240  177776  3$:     MOV     #<5*32.>,@#PS   ;RESTORE PRIORITY TO 5
2409
2410                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2411
2412  005174  005037  001250                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
2413  005200  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2414  005206  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
2415  005212  012737  011600  001124        MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
2416  005220  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2417  005226  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2418  005234  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
2419  005242  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2420  005250  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2421  005256  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2422  005264  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
2423  005272  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2424  005300  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2425  005306  001006                        BNE     66$             ;BR IF NOT
2426  005310  005737  001164                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2427  005314  001037                        BNE     68$             ;BR IF NOT
2428  005316  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
```

```
2429  005320  000137  005504                    JMP     70$             ;BYPASS THE REST OF THE CHECKS
2430  005324  013737  001170  001126  66$:      MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2431  005332  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2432  005340  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2433  005346  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2434  005352  001414                            BEQ     67$             ;BR IF ZERO
2435  005354  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2436  005362  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2437  005370  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2438  005376  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2439  005402  001004                            BNE     68$             ;BR IF NOT
2440  005404  012737  177777  001250  67$:      MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2441  005412  104022                            ERROR   22              ;TYPE ERROR MESSAGE 22
2442  005414  013737  001170  001126  68$:      MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
2443  005422  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
2444  005430  042737  100100  001170            BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2445  005436  023737  001124  001170            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
2446  005444  001401                            BEQ     69$             ;BR IF OK FROM PORT A.
2447  005446  104007                            ERROR   7               ;REPORT ERROR
2448  005450  013737  001172  001126  69$:      MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2449  005456  013737  001226  001234            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
2450  005464  042737  100100  001172            BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2451  005472  023737  001124  001172            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2452  005500  001401                            BEQ     70$             ;BR IF OK
2453  005502  104007                            ERROR   7               ;REPORT ERROR
2454  005504  000240                    70$:    NOP
2455
2456                                            ;;*************************************************************
2457                                            ;CHECK THE REGISTERS STORED THROUGH PORT B.  ALL REGISTERS SHOULD BE ZERO.
2458                                            ;THE REGISTERS ARE STORED ON THE STACK.
2459
2460  005506  013737  001226  001234  4$:       MOV     PORTB,PTNBR     ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2461  005514  010037  001122                    MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER1
2462  005520  062737  000014  001122            ADD     #RPER1,$BDADR   ;ADDRESS OF RPER1 FOR TYPEOUT
2463  005526  012637  001126                    MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER1
2464  005532  001401                            BEQ     .+4             ;CONTENTS ZERO ?
2465  005534  104006                            ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2466  005536  010037  001122                    MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPMR
2467  005542  062737  000024  001122            ADD     #RPMR,$BDADR    ;ADDRESS OF RPMR FOR TYPEOUT
2468  005550  012637  001126                    MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPMR
2469  005554  001401                            BEQ     .+4             ;CONTENTS ZERO ?
2470  005556  104006                            ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2471  005560  010037  001122                    MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDA
2472  005564  062737  000006  001122            ADD     #RPDA,$BDADR    ;ADDRESS OF RPDA FOR TYPEOUT
2473  005572  012637  001126                    MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDA
2474  005576  001401                            BEQ     .+4             ;CONTENTS ZERO ?
2475  005600  104006                            ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2476  005602  010037  001122                    MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDT
2477  005606  062737  000026  001122            ADD     #RPDT,$BDADR    ;ADDRESS OF RPDT FOR TYPEOUT
2478  005614  012637  001126                    MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDT
2479  005620  001401                            BEQ     .+4             ;CONTENTS ZERO ?
2480  005622  104006                            ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2481  005624  010037  001122                    MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPLA
2482  005630  062737  000020  001122            ADD     #RPLA,$BDADR    ;ADDRESS OF RPLA FOR TYPEOUT
2483  005636  012637  001126                    MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPLA
2484  005642  001401                            BEQ     .+4             ;CONTENTS ZERO ?
```

```
2485  005644  104006                        ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2486  005646  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER2
2487  005652  062737  000040  001122        ADD     #RPER2,$BDADR   ;ADDRESS OF RPER2 FOR TYPEOUT
2488  005660  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER2
2489  005664  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2490  005666  104006                        ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2491  005670  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPOF
2492  005674  062737  000032  001122        ADD     #RPOF,$BDADR    ;ADDRESS OF RPOF FOR TYPEOUT
2493  005702  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPOF
2494  005706  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2495  005710  104006                        ERROR   6               ;REPORT THAT PORT B SAW NON-ZERO REGISTER
2496  005712  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCA
2497  005716  062737  000034  001122        ADD     #RPCA,$BDADR    ;ADDRESS OF RPCA FOR TYPEOUT
2498  005724  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCA
2499  005730  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2500  005732  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2501  005734  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCC
2502  005740  062737  000036  001122        ADD     #RPCC,$BDADR    ;ADDRESS OF RPCC FOR TYPEOUT
2503  005746  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCC
2504  005752  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2505  005754  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2506  005756  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPSN
2507  005762  062737  000030  001122        ADD     #RPSN,$BDADR    ;ADDRESS OF RPSN FOR TYPEOUT
2508  005770  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPSN
2509  005774  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2510  005776  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2511  006000  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER3
2512  006004  062737  000042  001122        ADD     #RPER3,$BDADR   ;ADDRESS OF RPER3 FOR TYPEOUT
2513  006012  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER3
2514  006016  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2515  006020  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2516  006022  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPEC1
2517  006026  062737  000044  001122        ADD     #RPEC1,$BDADR   ;ADDRESS OF RPEC1 FOR TYPEOUT
2518  006034  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPEC1
2519  006040  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2520  006042  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2521  006044  010037  001122                MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPEC2
2522  006050  062737  000046  001122        ADD     #RPEC2,$BDADR   ;ADDRESS OF RPEC2 FOR TYPEOUT
2523  006056  012637  001126                MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPEC2
2524  006062  001401                        BEQ     .+4             ;CONTENTS ZERO ?
2525  006064  104006                        ERROR   6               ;REPORT THAT PORT B SEES NON-ZERO REGISTER
2526  006066  000004              5$:       SCOPE                   ;LOOP ?
2527
2528                                ;;*****************************************************************
2529                                ;*TEST 3        PORT 'B' SEIZE/TIMEOUT TEST
2530                                ;*
2531                                ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
2532                                ;*      THE DRIVE.
2533                                ;*
2534                                ;*  A.  WRITE 0'S INTO RPDS1 THROUGH PORT 'B';  VERIFY THAT THE DRIVE
2535                                ;*      HAS BEEN SEIZED.
2536                                ;*
2537                                ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'A';
2538                                ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
2539                                ;*
2540                                ;*  C.  WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
```

```
                                ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
                                ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
                                ;*      NEUTRAL.
                                ;*
                                ;;*****************************************************************
006070                          TST3:
006070  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
006074  001406                          BEQ     2$              ;BR IF NOT
006076  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
006100  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
006104  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
006112  112737  000003  001102  2$:     MOVB    #3,$TSTNM       ;TEST NUMBER
006120  012737  006142  001106          MOV     #TEST3,$LPADR   ;LOAD LOOP ON TEST ADDRESS
006126  012737  006142  001110          MOV     #TEST3,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
006134  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
006142  012706  001100          TEST3:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
006146  012737  000240  177776          MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
006154  005037  001264                  CLR     TIMEB           ;CLEAR TIMEOUT VALUE FOR PORT B
006160  005037  001266                  CLR     TIMEBP  ;CLEAR UPPER TIMEOUT TOLERANCE
006164  005037  001270                  CLR     TIMEBM  ;CLEAR LOWER TIMEOUT TOLERANCE

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

006170  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
006174  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS

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

                                        ;SEIZE THE DRIVE THROUGH PORT B

006202  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
006210  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
006216  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
006222  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
006230  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006236  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
006244  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
006252  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
006256  062737  000012  001122          ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
006264  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
006270  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
006276  001403                          BEQ     64$             ;BR IF IT IS
006300  104004                          ERROR   4               ;REPORT THE ERROR
006302  000137  007466                  JMP     5$              ;BYPASS REST OF THE SUBTEST
006306                          64$:
006306  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
006314  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006322  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
006330  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
006336  013737  001124  001166          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
006344  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
006350  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
006356  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
006364  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
006372  001401                          BEQ     65$             ;BR IF THEY ARE
```

```
                                        ERROR 5                 ;REPORT THE ERROR
006376  000240                  65$:   NOP

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

006400  113760  001224  000010         MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
006406  013737  001224  001234         MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006414  016046  000046               MOV     RPEC2(R0),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
006420  016046  000044               MOV     RPEC1(R0),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
006424  016046  000042               MOV     RPER3(R0),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
006430  016046  000030               MOV     RPSN(R0),-(SP)  ;STORE REGISTER RPSN, PORT A, FOR CHECK
006434  016046  000036               MOV     RPCC(R0),-(SP)  ;STORE REGISTER RPCC, PORT A, FOR CHECK
006440  016046  000034               MOV     RPCA(R0),-(SP)  ;STORE REGISTER RPCA, PORT A, FOR CHECK
006444  016046  000032               MOV     RPOF(R0),-(SP)  ;STORE REGISTER RPOF, PORT A, FOR CHECK
006450  016046  000040               MOV     RPER2(R0),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
006454  016046  000020               MOV     RPLA(R0),-(SP)  ;STORE REGISTER RPLA, PORT A, FOR CHECK
006460  016046  000026               MOV     RPDT(R0),-(SP)  ;STORE REGISTER RPDT, PORT A, FOR CHECK
006464  016046  000006               MOV     RPDA(R0),-(SP)  ;STORE REGISTER RPDA, PORT A, FOR CHECK
006470  016046  000024               MOV     RPMR(R0),-(SP)  ;STORE REGISTER RPMR, PORT A, FOR CHECK
006474  016046  000014               MOV     RPER1(R0),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK

                                ;;*****************************************************************
                                ;WAIT FOR PORT B TO TIMEOUT

006500  005760  000012         1$:   TST     RPDS1(R0)       ;WAIT FOR THE DRIVE TO TIMEOUT
006504  001006                       BNE     2$              ;BR WHEN TIMEOUT OCCURS
006506  005737  001254               TST     WATCH           ;CHECK WATCH
006512  001372                       BNE     1$              ;BR IF NOT ZERO
006514  104036                       ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
006516  000137  007106               JMP     4$              ;BYPASS TIMEOUT TIME CHECK
006522  012737  000340  177776 2$:   MOV     #<7*32.>,@#PS   ;SET PRIORITY TO 7 TO STOP CLOCK
006530  013737  001252  001264       MOV     TIME,TIMEB      ;SAVE THE ELAPSED TIME FOR PORT B
006536  004537  056630               JSR     R5,TOLER        ;CALCULATE THE TOLERANCE
006542  001264                       .WORD   TIMEB           ;TIMEOUT VALUE FOR PORT B
006544  012637  001266               MOV     (SP)+,TIMEBP    ;+25% TOLERANCE
006550  012637  001270               MOV     (SP)+,TIMEBM    ;-25% TOLERANCE

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

006554  023727  001252  000764         CMP     TIME,#500.      ;WAS MEASURED TIME AT LEAST 500 MS?
006562  103001                       BHIS    3$              ;BR IF IT  WAS
006564  104055                       ERROR   55              ;REPORT TIMEOUT TOO SHORT

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

006566  012737  000240  177776 3$:   MOV     #<5*32.>,@#PS   ;RESTORE PRIORITY TO 5

                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL

006574  005037  001250               CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
006600  012737  000012  001122       MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
006606  060037  001122               ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
006612  012737  011600  001124       MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
```

```
2653  006620  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2654  006626  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2655  006634  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
2656  006642  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2657  006650  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2658  006656  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2659  006664  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
2660  006672  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2661  006700  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2662  006706  001006                          BNE     66$             ;BR IF NOT
2663  006710  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2664  006714  001037                          BNE     68$             ;BR IF NOT
2665  006716  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2666  006720  000137  007104                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
2667  006724  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2668  006732  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2669  006740  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2670  006746  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2671  006752  001414                          BEQ     67$             ;BR IF ZERO
2672  006754  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2673  006762  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2674  006770  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2675  006776  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2676  007002  001004                          BNE     68$             ;BR IF NOT
2677  007004  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2678  007012  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
2679  007014  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
2680  007022  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
2681  007030  042737  100100  001170          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2682  007036  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
2683  007044  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
2684  007046  104007                          ERROR   7               ;REPORT ERROR
2685  007050  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2686  007056  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
2687  007064  042737  100100  001172          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2688  007072  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2689  007100  001401                          BEQ     70$             ;BR IF OK
2690  007102  104007                          ERROR   7               ;REPORT ERROR
2691  007104  000240                  70$:    NOP
2692
2693                                          ;;*****************************************************************
2694                                          ;CHECK THE REGISTERS STORED THROUGH PORT A.  ALL REGISTERS SHOULD BE ZERO.
2695                                          ;THE REGISTERS ARE STORED ON THE STACK.
2696
2697  007106  013737  001224  001234  4$:     MOV     PORTA,PTNBR     ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
2698  007114  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER1
2699  007120  062737  000014  001122          ADD     #RPER1,$BDADR   ;ADDRESS OF RPER1 FOR TYPEOUT
2700  007126  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER1
2701  007132  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2702  007134  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2703  007136  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPMR
2704  007142  062737  000024  001122          ADD     #RPMR,$BDADR    ;ADDRESS OF RPMR FOR TYPEOUT
2705  007150  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPMR
2706  007154  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2707  007156  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2708  007160  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDA
```

```
2709  007164  062737  000006  001122          ADD     #RPDA,$BDADR    ;ADDRESS OF RPDA FOR TYPEOUT
2710  007172  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDA
2711  007176  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2712  007200  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2713  007202  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDT
2714  007206  062737  000026  001122          ADD     #RPDT,$BDADR    ;ADDRESS OF RPDT FOR TYPEOUT
2715  007214  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDT
2716  007220  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2717  007222  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2718  007224  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPLA
2719  007230  062737  000020  001122          ADD     #RPLA,$BDADR    ;ADDRESS OF RPLA FOR TYPEOUT
2720  007236  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPLA
2721  007242  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2722  007244  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2723  007246  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER2
2724  007252  062737  000040  001122          ADD     #RPER2,$BDADR   ;ADDRESS OF RPER2 FOR TYPEOUT
2725  007260  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER2
2726  007264  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2727  007266  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2728  007270  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPOF
2729  007274  062737  000032  001122          ADD     #RPOF,$BDADR    ;ADDRESS OF RPOF FOR TYPEOUT
2730  007302  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPOF
2731  007306  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2732  007310  104006                          ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
2733  007312  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCA
2734  007316  062737  000034  001122          ADD     #RPCA,$BDADR    ;ADDRESS OF RPCA FOR TYPEOUT
2735  007324  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCA
2736  007330  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2737  007332  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2738  007334  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCC
2739  007340  062737  000036  001122          ADD     #RPCC,$BDADR    ;ADDRESS OF RPCC FOR TYPEOUT
2740  007346  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCC
2741  007352  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2742  007354  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2743  007356  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPSN
2744  007362  062737  000030  001122          ADD     #RPSN,$BDADR    ;ADDRESS OF RPSN FOR TYPEOUT
2745  007370  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPSN
2746  007374  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2747  007376  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2748  007400  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER3
2749  007404  062737  000042  001122          ADD     #RPER3,$BDADR   ;ADDRESS OF RPER3 FOR TYPEOUT
2750  007412  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER3
2751  007416  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2752  007420  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2753  007422  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPEC1
2754  007426  062737  000044  001122          ADD     #RPEC1,$BDADR   ;ADDRESS OF RPEC1 FOR TYPEOUT
2755  007434  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPEC1
2756  007440  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2757  007442  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2758  007444  010037  001122                  MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPEC2
2759  007450  062737  000046  001122          ADD     #RPEC2,$BDADR   ;ADDRESS OF RPEC2 FOR TYPEOUT
2760  007456  012637  001126                  MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPEC2
2761  007462  001401                          BEQ     .+4             ;CONTENTS ZERO ?
2762  007464  104006                          ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
2763  007466  000004                  5$:     SCOPE                   ;LOOP ?
2764
```

```
                        ;;*****************************************************************
                        ;*TEST 4        PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
                        ;*
                        ;*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
                        ;*      FOR THE PORT UNDER TEST.
                        ;*
                        ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
                        ;*      DRIVE WAS SEIZIED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
                        ;*
                        ;*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
                        ;*      'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
                        ;*      FOR PORT 'B'.  (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
                        ;*      'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
                        ;*
                        ;*  C.  STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
                        ;*      THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
                        ;*
                        ;;*****************************************************************
007470                  TST4:
007470  005737  001274          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
007474  001406                  BEQ     2$              ;BR IF NOT
007476  100002                  BPL     1$              ;BR IF JUST ENTERED TEST
007500  000137  002602          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
007504  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
007512  112737  000004  001102  2$:     MOVB    #4,$TSTNM       ;TEST NUMBER
007520  012737  007542  001106          MOV     #TEST4,$LPADR   ;LOAD LOOP ON TEST ADDRESS
007526  012737  007542  001110          MOV     #TEST4,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
007534  012737  000001  001176          MOV     #1,$TIMES       ;;DO 1 ITERATION
007542  012706  001100          TEST4:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
007546  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
007554  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

007562  005037  001252          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
007566  012737  003720  001254  MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
007574  013737  001224  001236  MOV     PORTA,$EIZPT    ;'SEIZED' PORT ADDRESS

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

007602  012760  000011  000000  MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR

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

007610  113760  001226  000010  MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
007616  013737  001226  001234  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
007624  005037  001244          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
007630  016037  000012  001126  MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
007636  012737  000012  001122  MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
007644  060037  001122          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
007650  005037  001124          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
007654  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
```

```
2821  007662  001403                        BEQ     64$             ;BR IF OK
2822  007664  104012                        ERROR   12              ;TYPE MESSAGE 12
2823  007666  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2824  007672  000240                64$:    NOP
2825  007674  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2826  007702  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2827  007710  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2828  007714  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2829  007722  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2830  007730  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2831  007734  012737  011600  001124        MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
2832  007742  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2833  007750  042737  106177  001164        BIC     #^C71600,$TMP0  ;SAVE SPECIFIED BITS
2834  007756  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2835  007764  001414                        BEQ     66$             ;BR IF OK
2836  007766  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2837  007774  042737  071600  001174        BIC     #71600,$TMP4    ;CLEAR THE MASKED BITS
2838  010002  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2839  010010  104010                        ERROR   10              ;REPORT THE ERROR
2840  010012  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2841  010016  000240                66$:    NOP
2842  010020  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2843  010024  016037  000000  001126        MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
2844  010032  012737  000000  001122        MOV     #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2845  010040  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2846  010044  012737  004210  001124        MOV     #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
2847  010052  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2848  010060  042737  100000  001164        BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
2849  010066  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2850  010074  001414                        BEQ     68$             ;BR IF OK
2851  010076  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
2852  010104  042737  077777  001174        BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
2853  010112  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2854  010120  104010                        ERROR   10              ;REPORT THE ERROR
2855  010122  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2856  010126  000240                68$:    NOP
2857
2858                                  ;;*****************************************************************
2859                                  ;ISSUE READIN PRESET COMMAND AND SET FMT22
2860
2861  010130  012760  000023  000000        MOV     #23,RPCS1(R0)  ;ISSUE A READIN PRESET
2862  010136  012760  010000  000032        MOV     #FMT22,RPOF(R0) ;SET FMT22
2863
2864                                  ;;*****************************************************************
2865                                  ;VERIFY THAT THE DRIVE STATUS IS CORRECT
2866
2867  010144  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2868  010150  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2869  010156  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2870  010164  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2871  010170  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
2872  010176  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
2873  010204  042737  106077  001164        BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
2874  010212  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
2875  010220  001414                        BEQ     70$             ;BR IF OK
2876  010222  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
```

```
2877  010230  042737  071700  001174          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
2878  010236  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
2879  010244  104013                          ERROR   13              ;TYPE MESSAGE 13
2880  010246  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
2881  010252  000240                  70$:    NOP
2882  010254  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
2883  010262  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2884
2885                                  ;;*****************************************************************
2886                                  ;WAIT FOR TIMEOUT TO RELEASE DRIVE
2887
2888  010270  005760  000012          1$:     TST     RPDS1(R0)       ;WAIT FOR THE PORT TO TIME OUT
2889  010274  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
2890  010276  005737  001254                  TST     WATCH           ;CHECK THE WATCH
2891  010302  001372                          BNE     1$              ;BR IF NOT ZERO
2892  010304  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
2893  010306  000137  010624                  JMP     3$              ;BYPASS ATTN REGISTER CHECK
2894
2895                                  ;;*****************************************************************
2896                                  ;SEE IF DRIVE RETURNED TO NEUTRAL
2897
2898  010312                          2$:
2899
2900                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2901
2902  010312  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
2903  010316  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
2904  010324  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
2905  010330  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
2906  010336  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2907  010344  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2908  010352  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
2909  010360  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2910  010366  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2911  010374  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2912  010402  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
2913  010410  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2914  010416  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2915  010424  001006                          BNE     72$             ;BR IF NOT
2916  010426  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2917  010432  001037                          BNE     74$             ;BR IF NOT
2918  010434  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2919  010436  000137  010622                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
2920  010442  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2921  010450  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2922  010456  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
2923  010464  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
2924  010470  001414                          BEQ     73$             ;BR IF ZERO
2925  010472  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2926  010500  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
2927  010506  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
2928  010514  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
2929  010520  001004                          BNE     74$             ;BR IF NOT
2930  010522  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
2931  010530  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
2932  010532  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
```

```
2933  010540  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
2934  010546  042737  100100  001170          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
2935  010554  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
2936  010562  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
2937  010564  104007                          ERROR   7               ;REPORT ERROR
2938  010566  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
2939  010574  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
2940  010602  042737  100100  001172          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
2941  010610  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
2942  010616  001401                          BEQ     76$             ;BR IF OK
2943  010620  104007                          ERROR   7               ;REPORT ERROR
2944  010622  000240                  76$:    NOP
2945  010624  000004                  3$:     SCOPE                   ;LOOP ?
2946
2947                                  ;;*****************************************************************
2948                                  ;*TEST 5        PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'
2949                                  ;*
2950                                  ;*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
2951                                  ;*      FOR THE PORT UNDER TEST.
2952                                  ;*
2953                                  ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
2954                                  ;*      DRIVE WAS SEIZIED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
2955                                  ;*
2956                                  ;*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'.  VERIFY THAT THE
2957                                  ;*      'VV' BIT FOR PORT 'B' WAS SET.
2958                                  ;*
2959                                  ;*  C.  STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
2960                                  ;*      THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
2961                                  ;*
2962                                  ;;*****************************************************************
2963  010626                          TST5:
2964  010626  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2965  010632  001406                          BEQ     2$              ;BR IF NOT
2966  010634  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
2967  010636  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
2968  010642  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
2969  010650  112737  000005  001102  2$:     MOVB    #5,$TSTNM       ;TEST NUMBER
2970  010656  012737  010700  001106          MOV     #TEST5,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2971  010664  012737  010700  001110          MOV     #TEST5,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
2972  010672  012737  000001  001176          MOV     #1,$TIMES       ;;DO 1 ITERATION
2973  010700  012706  001100          TEST5:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2974  010704  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
2975  010712  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2976
2977                                  ;;*****************************************************************
2978                                  ;START THE TIMER
2979
2980  010720  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
2981  010724  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
2982  010732  013737  001226  001236          MOV     PORTB,$EIZPT    ;'SEIZED' PORT ADDRESS
2983
2984                                  ;;*****************************************************************
2985                                  ;ISSUE DRIVE CLEAR COMMAND
2986
2987  010740  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
2988
```

```
2989                                    ;;*****************************************************************
2990                                    ;VERIFY THAT DRIVE SEIZED BY PORT B.
2991
2992  010746  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
2993  010754  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2994  010762  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
2995  010766  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2996  010774  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2997  011002  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
2998  011006  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
2999  011012  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
3000  011020  001403                          BEQ     64$             ;BR IF OK
3001  011022  104012                          ERROR   12              ;TYPE MESSAGE 12
3002  011024  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3003  011030  000240                  64$:    NOP
3004  011032  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3005  011040  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3006  011046  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3007  011052  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3008  011060  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3009  011066  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3010  011072  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;WHAT REGISTER SHOULD BE
3011  011100  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3012  011106  042737  106177  001164          BIC     #^C71600,$TMP0  ;SAVE SPECIFIED BITS
3013  011114  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3014  011122  001414                          BEQ     66$             ;BR IF OK
3015  011124  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3016  011132  042737  071600  001174          BIC     #71600,$TMP4    ;CLEAR THE MASKED BITS
3017  011140  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3018  011146  104010                          ERROR   10              ;REPORT THE ERROR
3019  011150  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3020  011154  000240                  66$:    NOP
3021  011156  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3022  011162  016037  000000  001126          MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
3023  011170  012737  000000  001122          MOV     #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3024  011176  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3025  011202  012737  004210  001124          MOV     #4210,$GDDAT  ;WHAT REGISTER SHOULD BE
3026  011210  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3027  011216  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3028  011224  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3029  011232  001414                          BEQ     68$             ;BR IF OK
3030  011234  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3031  011242  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3032  011250  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3033  011256  104010                          ERROR   10              ;REPORT THE ERROR
3034  011260  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3035  011264  000240                  68$:    NOP
3036
3037                                    ;;*****************************************************************
3038                                    ;ISSUE READIN PRESET COMMAND AND SET FMT22
3039
3040  011266  012760  000023  000000          MOV     #23,RPCS1(R0)  ;ISSUE A READIN PRESET
3041  011274  012760  010000  000032          MOV     #FMT22,RPOF(R0) ;SET FMT22
3042
3043                                    ;;*****************************************************************
3044                                    ;VERIFY THAT THE DRIVE STATUS IS CORRECT
```

```
3045
3046  011302  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3047  011306  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3048  011314  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3049  011322  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3050  011326  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3051  011334  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3052  011342  042737  106077  001164          BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
3053  011350  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3054  011356  001414                          BEQ     70$             ;BR IF OK
3055  011360  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3056  011366  042737  071700  001174          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
3057  011374  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3058  011402  104013                          ERROR   13              ;TYPE MESSAGE 13
3059  011404  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3060  011410  000240                  70$:    NOP
3061  011412  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3062  011420  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3063
3064                                  ;;*****************************************************************
3065                                  ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3066
3067  011426  005760  000012          1$:     TST     RPDS1(R0)       ;WAIT FOR THE PORT TO TIME OUT
3068  011432  001006                          BNE     2$              ;BR WHEN TIMEOUT OCCURS
3069  011434  005737  001254                  TST     WATCH           ;CHECK THE WATCH
3070  011440  001372                          BNE     1$              ;BR IF NOT ZERO
3071  011442  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3072  011444  000137  011762                  JMP     3$              ;BYPASS ATTN REGISTER CHECK
3073
3074                                  ;;*****************************************************************
3075                                  ;SEE IF DRIVE RETURNED TO NEUTRAL
3076
3077  011450                          2$:
3078
3079                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3080
3081  011450  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3082  011454  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3083  011462  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3084  011466  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3085  011474  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3086  011502  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3087  011510  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3088  011516  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3089  011524  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3090  011532  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3091  011540  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3092  011546  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3093  011554  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3094  011562  001006                          BNE     72$             ;BR IF NOT
3095  011564  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3096  011570  001037                          BNE     74$             ;BR IF NOT
3097  011572  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3098  011574  000137  011760                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
3099  011600  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3100  011606  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
3101  011614  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3102  011622  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3103  011626  001414                          BEQ     73$             ;BR IF ZERO
3104  011630  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3105  011636  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3106  011644  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3107  011652  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3108  011656  001004                          BNE     74$             ;BR IF NOT
3109  011660  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3110  011666  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3111  011670  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3112  011676  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3113  011704  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3114  011712  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3115  011720  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
3116  011722  104007                          ERROR   7               ;REPORT ERROR
3117  011724  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3118  011732  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3119  011740  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3120  011746  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3121  011754  001401                          BEQ     76$             ;BR IF OK
3122  011756  104007                          ERROR   7               ;REPORT ERROR
3123  011760  000240                  76$:    NOP
3124  011762  000004                  3$:     SCOPE                   ;LOOP ?
3125
3126
3127                                          ;;*****************************************************************
3128                                          ;*TEST 6        TEST RELEASE, DRIVE SEIZED BY PORT 'A'
3129                                          ;*
3130                                          ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
3131                                          ;*
3132                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
3133                                          ;*
3134                                          ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT'A'.  VERIFY THAT THE DRIVE
3135                                          ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
3136                                          ;*      DRIVE.
3137                                          ;*
3138                                          ;;*****************************************************************
3139  011764                                  TST6:
3140  011764  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3141  011770  001406                          BEQ     2$              ;BR IF NOT
3142  011772  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3143  011774  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3144  012000  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3145  012006  112737  000006  001102  2$:     MOVB    #6,$TSTNM       ;TEST NUMBER
3146  012014  012737  012036  001106          MOV     #TEST6,$LPADR   ;LOAD LOOP ON TEST ADDRESS
3147  012022  012737  012036  001110          MOV     #TEST6,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
3148  012030  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
3149  012036  012706  001100          TEST6:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3150
3151                                          ;;*****************************************************************
3152                                          ;START THE TIMER
3153
3154  012042  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
3155  012046  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
3156
```

```
                                        ;;*****************************************************************
                                                ;SEIZE THE DRIVE THROUGH PORT A
012054  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
012062  013737  001224  001236                  MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
012070  005060  000012                          CLR     RPDS1(R0)       ;WRITE RPDS1
012074  013737  001226  001240                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
                                        ;;*****************************************************************
                                                ;RELEASE THE DRIVE FROM PORT A
012102  113760  001224  000010                  MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
012110  013737  001224  001234                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012116  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
012124  005037  001250                          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
012130  012737  000012  001122                  MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
012136  060037  001122                          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
012142  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
012150  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
012156  016037  000012  001170                  MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012164  013737  001170  001164                  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
012172  042737  100100  001164                  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012200  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
012206  016037  000012  001172                  MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
012214  013737  001172  001166                  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
012222  042737  100100  001166                  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012230  023737  001164  001166                  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
012236  001006                                  BNE     66$             ;BR IF NOT
012240  005737  001164                          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
012244  001037                                  BNE     68$             ;BR IF NOT
012246  104046                                  ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
012250  000137  012434                          JMP     70$             ;BYPASS THE REST OF THE CHECKS
012254  013737  001170  001126  66$:            MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012262  013737  001226  001234                  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012270  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
012276  005737  001164                          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
012302  001414                                  BEQ     67$             ;BR IF ZERO
012304  013737  001224  001234                  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012312  013737  001172  001126                  MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
012320  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
012326  005737  001166                          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
012332  001004                                  BNE     68$             ;BR IF NOT
012334  012737  177777  001250  67$:            MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
012342  104026                                  ERROR   26              ;TYPE ERROR MESSAGE 26
012344  013737  001170  001126  68$:            MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
012352  013737  001224  001234                  MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
012360  042737  100000  001170                  BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
012366  023737  001124  001170                  CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
012374  001401                                  BEQ     69$             ;BR IF OK FROM PORT A.
012376  104007                                  ERROR   7               ;REPORT ERROR
012400  013737  001172  001126  69$:            MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
```

```
3213  012406  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3214  012414  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3215  012422  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3216  012430  001401                          BEQ     70$             ;BR IF OK
3217  012432  104007                          ERROR   7               ;REPORT ERROR
3218  012434  000240                  70$:    NOP
3219  012436  005737  001250                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
3220  012442  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
3221  012444  000137  012720                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
3222  012450  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
3223  012456  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3224  012464  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3225  012470  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3226  012476  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3227  012504  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3228  012510  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3229  012514  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3230  012522  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
3231  012530  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3232  012536  001414                          BEQ     71$             ;BR IF OK
3233  012540  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3234  012546  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3235  012554  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3236  012562  104017                          ERROR   17              ;TYPE MESSAGE 17
3237  012564  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3238  012570  000240                  71$:    NOP
3239  012572  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
3240  012600  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3241  012606  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3242  012612  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3243  012620  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3244  012626  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3245  012632  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3246  012636  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3247  012644  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
3248  012652  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3249  012660  001414                          BEQ     73$             ;BR IF OK
3250  012662  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3251  012670  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3252  012676  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3253  012704  104017                          ERROR   17              ;TYPE MESSAGE 17
3254  012706  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3255  012712  000240                  73$:    NOP
3256  012714  000137  012752                  JMP     2$              ;GO CHECK FOR LOOP ON ERROR
3257
3258                                  ;;*****************************************************************
3259                                  ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3260                                  ;TO RELEASE THE DRIVE
3261
3262  012720                          1$:
3263  012720  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
3264  012726  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3265  012734  005760  000012                  TST     RPDS1(R0)       ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3266  012740  001004                          BNE     2$              ;BR WHEN DRIVE RELEASED
3267  012742  005737  001254                  TST     WATCH           ;CHECK THE WATCH
3268  012746  001364                          BNE     1$              ;BR IF NOT ZERO
```

```
                                        T6      TEST RELEASE, DRIVE SEIZED BY PORT 'A'

 3269  012750  104036                           ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
 3270  012752  000004                 2$:       SCOPE                   ;LOOP ?
 3271
 3272                                           ;;*****************************************************************
 3273                                           ;*TEST 7        TEST RELEASE, DRIVE SEIZED BY PORT 'B'
 3274                                           ;*
 3275                                           ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
 3276                                           ;*
 3277                                           ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 3278                                           ;*
 3279                                           ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
 3280                                           ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
 3281                                           ;*      DRIVE.
 3282                                           ;*
 3283                                           ;;*****************************************************************
 3284  012754                                   TST7:
 3285  012754  005737  001274                           TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 3286  012760  001406                           2$      BEQ     2$              ;BR IF NOT
 3287  012762  100002                           1$      BPL     1$              ;BR IF JUST ENTERED TEST
 3288  012764  000137  002602                           JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 3289  012770  012737  177777  001274   1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 3290  012776  112737  000007  001102   2$:     MOVB    #7,$TSTNM       ;TEST NUMBER
 3291  013004  012737  013026  001106           MOV     #TEST7,$LPADR   ;LOAD LOOP ON TEST ADDRESS
 3292  013012  012737  013026  001110           MOV     #TEST7,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
 3293  013020  012737  007640  001176           MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 3294  013026  012706  001100           TEST7:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 3295
 3296                                           ;;*****************************************************************
 3297                                           ;START THE TIMER
 3298
 3299  013032  005037  001252                           CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
 3300  013036  012737  003720  001254           MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
 3301
 3302                                           ;;*****************************************************************
 3303
 3304                                           ;SEIZE THE DRIVE THROUGH PORT B
 3305
 3306  013044  113760  001226  000010           MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 3307  013052  013737  001226  001236           MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 3308  013060  005060  000012                   CLR     RPDS1(R0)       ;WRITE RPDS1
 3309  013064  013737  001224  001240           MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
 3310
 3311                                           ;;*****************************************************************
 3312
 3313                                           ;RELEASE THE DRIVE FROM PORT B
 3314
 3315  013072  113760  001226  000010           MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 3316  013100  013737  001226  001234           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 3317  013106  012760  000013  000000           MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
 3318
 3319                                           ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 3320
 3321  013114  005037  001250                   CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 3322  013120  012737  000012  001122           MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 3323  013126  060037  001122                   ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 3324  013132  012737  011700  001124           MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
```

```
3325  013140  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3326  013146  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3327  013154  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3328  013162  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3329  013170  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3330  013176  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3331  013204  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3332  013212  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3333  013220  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3334  013226  001006                          BNE     66$             ;BR IF NOT
3335  013230  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3336  013234  001037                          BNE     68$             ;BR IF NOT
3337  013236  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3338  013240  000137  013424                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
3339  013244  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3340  013252  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3341  013260  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3342  013266  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3343  013272  001414                          BEQ     67$             ;BR IF ZERO
3344  013274  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3345  013302  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3346  013310  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3347  013316  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3348  013322  001004                          BNE     68$             ;BR IF NOT
3349  013324  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3350  013332  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3351  013334  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3352  013342  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3353  013350  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3354  013356  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3355  013364  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
3356  013366  104007                          ERROR   7               ;REPORT ERROR
3357  013370  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3358  013376  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3359  013404  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3360  013412  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3361  013420  001401                          BEQ     70$             ;BR IF OK
3362  013422  104007                          ERROR   7               ;REPORT ERROR
3363  013424  000240                  70$:    NOP
3364  013426  005737  001250                  TST     RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
3365  013432  001402                          BEQ     .+6             ;BR IF IN NEUTRAL
3366  013434  000137  013710                  JMP     1$              ;GO WAIT FOR DRIVE TO TIMEOUT
3367  013440  113760  001226  000010          MOVB-   PORTB,RPCS2(R0) ;SELECT PORT B
3368  013446  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3369  013454  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3370  013460  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3371  013466  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3372  013474  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3373  013500  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3374  013504  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3375  013512  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
3376  013520  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3377  013526  001414                          BEQ     71$             ;BR IF OK
3378  013530  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3379  013536  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3380  013544  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
3381  013552  104017                                ERROR   17              ;TYPE MESSAGE 17
3382  013554  005137  001244                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3383  013560  000240                        71$:    NOP
3384  013562  113760  001224  000010                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3385  013570  013737  001224  001234                MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3386  013576  005037  001244                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3387  013602  016037  000012  001126                MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3388  013610  012737  000012  001122                MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3389  013616  060037  001122                        ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3390  013622  005037  001124                        CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3391  013626  013737  001126  001164                MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3392  013634  042737  077777  001164                BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
3393  013642  023737  001124  001164                CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3394  013650  001414                                BEQ     73$             ;BR IF OK
3395  013652  013737  001126  001174                MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3396  013660  042737  100000  001174                BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
3397  013666  053737  001174  001124                BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3398  013674  104017                                ERROR   17              ;TYPE MESSAGE 17
3399  013676  005137  001244                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3400  013702  000240                        73$:    NOP
3401  013704  000137  013742                        JMP     2$              ;GO CHECK FOR LOOP ON ERROR
3402
3403                                        ;;**************************************************************
3404                                        ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
3405                                        ;TO RELEASE THE DRIVE
3406
3407  013710                                1$:
3408  013710  113760  001224  000010                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3409  013716  013737  001224  001234                MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3410  013724  005760  000012                        TST     RPDS1(R0)       ;WAIT FOR TIMEOUT TO RELEASE DRIVE
3411  013730  001004                                BNE     2$              ;BR WHEN DRIVE RELEASED
3412  013732  005737  001254                        TST     WATCH           ;CHECK THE WATCH
3413  013736  001364                                BNE     1$              ;BR IF NOT ZERO
3414  013740  104036                                ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3415  013742  000004                        2$:     SCOPE                   ;LOOP ?
3416
3417
3418                                        ;;**************************************************************
3419                                        ;*TEST 10       TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL
3420                                        ;*
3421                                        ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
3422                                        ;*
3423                                        ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN
3424                                        ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
3425                                        ;*
3426                                        ;;**************************************************************
3427  013744                                TST10:
3428  013744  005737  001274                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3429  013750  001406                                BEQ     2$              ;BR IF NOT
3430  013752  100002                                BPL     1$              ;BR IF JUST ENTERED TEST
3431  013754  000137  002602                        JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3432  013760  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3433  013766  112737  000010  001102  2$:     MOVB    #10,$TSTNM      ;TEST NUMBER
3434  013774  012737  014016  001106                MOV     #TEST10,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3435  014002  012737  014016  001110                MOV     #TEST10,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3436  014010  012737  000144  001176                MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
```

```
3437  014016  012706  001100          TEST10: MOV     #STACK,SP       ;LOAD THE STACK POINTER
3438  014022  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
3439  014030  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3440  014036  013737  001224  001236          MOV     PORTA,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3441
3442                                          ;;**************************************************************
3443                                          ;ISSUE A RELEASE COMMAND
3444
3445  014044  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
3446
3447                                          ;;**************************************************************
3448                                          ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
3449
3450
3451                                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3452
3453  014052  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3454  014056  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3455  014064  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3456  014070  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3457  014076  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3458  014104  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3459  014112  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3460  014120  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3461  014126  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3462  014134  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3463  014142  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3464  014150  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3465  014156  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3466  014164  001006                          BNE     64$             ;BR IF NOT
3467  014166  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3468  014172  001037                          BNE     66$             ;BR IF NOT
3469  014174  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3470  014176  000137  014362                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
3471  014202  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3472  014210  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3473  014216  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3474  014224  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3475  014230  001414                          BEQ     65$             ;BR IF ZERO
3476  014232  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3477  014240  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3478  014246  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3479  014254  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3480  014260  001004                          BNE     66$             ;BR IF NOT
3481  014262  012737  177777  001250  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3482  014270  104030                          ERROR   30              ;TYPE ERROR MESSAGE 30
3483  014272  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3484  014300  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3485  014306  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3486  014314  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3487  014322  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3488  014324  104007                          ERROR   7               ;REPORT ERROR
3489  014326  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3490  014334  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3491  014342  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3492  014350  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
```

```
3493  014356  001401                          BEQ     68$             ;BR IF OK
3494  014360  104007                          ERROR   7               ;REPORT ERROR
3495  014362  000240                  68$:    NOP
3496  014364  000004                          SCOPE                   ;LOOP ?
3497
3498                                  ;;**********************************************************************
3499                                  ;*TEST 11       TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
3500                                  ;*
3501                                  ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
3502                                  ;*
3503                                  ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
3504                                  ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
3505                                  ;*
3506                                  ;;**********************************************************************
3507  014366                          TST11:
3508  014366  005737  001274                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3509  014372  001406                         BEQ     2$              ;BR IF NOT
3510  014374  100002                         BPL     1$              ;BR IF JUST ENTERED TEST
3511  014376  000137  002602                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3512  014402  012737  177777  001274  1$:    MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3513  014410  112737  000011  001102  2$:    MOVB    #11,$TSTNM      ;TEST NUMBER
3514  014416  012737  014440  001106         MOV     #TEST11,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3515  014424  012737  014440  001110         MOV     #TEST11,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3516  014432  012737  000144  001176         MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3517  014440  012706  001100          TEST11: MOV     #STACK,SP       ;LOAD THE STACK POINTER
3518  014444  113760  001226  000010         MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3519  014452  013737  001226  001234         MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3520  014460  013737  001226  001236         MOV     PORTB,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE
3521
3522                                  ;;**********************************************************************
3523                                  ;ISSUE A RELEASE COMMAND
3524
3525  014466  012760  000013  000000         MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
3526
3527                                  ;;**********************************************************************
3528                                  ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
3529
3530
3531                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3532
3533  014474  005037  001250                 CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3534  014500  012737  000012  001122         MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3535  014506  060037  001122                 ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3536  014512  012737  011700  001124         MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3537  014520  113760  001224  000010         MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3538  014526  016037  000012  001170         MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3539  014534  013737  001170  001164         MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3540  014542  042737  100100  001164         BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3541  014550  113760  001226  000010         MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3542  014556  016037  000012  001172         MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3543  014564  013737  001172  001166         MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3544  014572  042737  100100  001166         BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3545  014600  023737  001164  001166         CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3546  014606  001006                         BNE     64$             ;BR IF NOT
3547  014610  005737  001164                 TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3548  014614  001037                         BNE     66$             ;BR IF NOT
```

```
3549  014616  104046                                ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3550  014620  000137  015004                        JMP     68$             ;BYPASS THE REST OF THE CHECKS
3551  014624  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3552  014632  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3553  014640  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3554  014646  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3555  014652  001414                          BEQ     65$             ;BR IF ZERO
3556  014654  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3557  014662  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3558  014670  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3559  014676  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3560  014702  001004                          BNE     66$             ;BR IF NOT
3561  014704  012737  177777  001250  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3562  014712  104030                          ERROR   30              ;TYPE ERROR MESSAGE 30
3563  014714  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3564  014722  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3565  014730  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3566  014736  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3567  014744  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
3568  014746  104007                          ERROR   7               ;REPORT ERROR
3569  014750  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3570  014756  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3571  014764  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3572  014772  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3573  015000  001401                          BEQ     68$             ;BR IF OK
3574  015002  104007                          ERROR   7               ;REPORT ERROR
3575  015004  000240                  68$:    NOP
3576  015006  000004                          SCOPE                   ;LOOP ?
3577
3578
3579                                  ;;*****************************************************************
3580                                  ;*TEST 12       TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'
3581                                  ;*
3582                                  ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3583                                  ;*      PORT TO RELEASE THE DRIVE.
3584                                  ;*
3585                                  ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'A'.
3586                                  ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
3587                                  ;*
3588                                  ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
3589                                  ;*      DOES NOT RETURN TO NEUTRAL.
3590                                  ;*
3591                                  ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
3592                                  ;*      DOES NOT RETURN TO NEUTRAL.
3593                                  ;*
3594                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
3595                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3596                                  ;*
3597                                  ;;*****************************************************************
3598  015010                          TST12:
3599  015010  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3600  015014  001406                          BEQ     2$              ;BR IF NOT
3601  015016  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
3602  015020  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
3603  015024  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
3604  015032  112737  000012  001102  2$:     MOVB    #12,$TSTNM      ;TEST NUMBER
```

```
                                T12     TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'

 3605  015040  012737  015062  001106         MOV     #TEST12,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 3606  015046  012737  015062  001110         MOV     #TEST12,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 3607  015054  012737  007640  001176         MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 3608  015062  012706  001100         TEST12: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 3609
 3610                                 ;;**************************************************************
 3611
 3612                                         ;SEIZE THE DRIVE THROUGH PORT A
 3613
 3614  015066  113760  001224  000010         MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 3615  015074  013737  001224  001236         MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
 3616  015102  005060  000012                 CLR     RPDS1(R0)       ;WRITE RPDS1
 3617  015106  113760  001226  000010         MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 3618  015114  013737  001226  001234         MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 3619  015122  013737  001226  001240         MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
 3620  015130  016037  000012  001126         MOV     RPDS1(R0),$BDDAT  ;SEE IF DRIVE SEIZED BY PORT A
 3621  015136  010037  001122                 MOV     R0,$BDADR       ;RH11 BASE ADDRESS
 3622  015142  062737  000012  001122         ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
 3623  015150  005037  001124                 CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
 3624  015154  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
 3625  015162  001403                         BEQ     64$             ;BR IF IT IS
 3626  015164  104004                         ERROR   4               ;REPORT THE ERROR
 3627  015166  000137  016364                 JMP     1$              ;BYPASS REST OF THE SUBTEST
 3628  015172                         64$:
 3629  015172  113760  001224  000010         MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 3630  015200  013737  001224  001234         MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 3631  015206  016037  000012  001126         MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
 3632  015214  012737  011700  001124         MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
 3633  015222  013737  001124  001166         MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
 3634  015230  005137  001166                 COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
 3635  015234  013737  001126  001164         MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
 3636  015242  043737  001166  001164         BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
 3637  015250  023737  001124  001164         CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
 3638  015256  001401                         BEQ     65$             ;BR IF THEY ARE
 3639  015260  104005                         ERROR 5                 ;REPORT THE ERROR
 3640  015262  000240                 65$:    NOP
 3641
 3642                                 ;;**************************************************************
 3643                                 ;DRIVE CLEAR THROUGH PORT A FIRST
 3644
 3645  015264  012760  000011  000000         MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT A
 3646
 3647                                 ;;**************************************************************
 3648                                 ;VERIFY THAT DRIVE STILL SEIZED BY PORT A
 3649
 3650  015272  113760  001226  000010         MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 3651  015300  013737  001226  001234         MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 3652  015306  005037  001244                 CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 3653  015312  016037  000012  001126         MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 3654  015320  012737  000012  001122         MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 3655  015326  060037  001122                 ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 3656  015332  005037  001124                 CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 3657  015336  013737  001126  001164         MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
 3658  015344  042737  100000  001164         BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
 3659  015352  023737  001124  001164         CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
 3660  015360  001414                         BEQ     66$             ;BR IF OK
```

```
3661  015362  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3662  015370  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3663  015376  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3664  015404  104033                          ERROR   33              ;TYPE MESSAGE 33
3665  015406  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3666  015412  000240                  66$:    NOP
3667  015414  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3668  015422  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3669  015430  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3670  015434  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3671  015442  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3672  015450  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3673  015454  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3674  015462  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3675  015470  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3676  015476  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3677  015504  001414                          BEQ     68$             ;BR IF OK
3678  015506  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3679  015514  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3680  015522  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3681  015530  104033                          ERROR   33              ;TYPE MESSAGE 33
3682  015532  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3683  015536  000240                  68$:    NOP
3684
3685                                  ;;*****************************************************************
3686                                  ;NOW ISSUE MASSBUS INIT
3687
3688  015540  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;ISSUE MASSBUS INIT
3689
3690                                  ;;*****************************************************************
3691                                  ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A
3692
3693  015546  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3694  015554  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3695  015562  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3696  015566  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3697  015574  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3698  015602  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3699  015606  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3700  015612  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3701  015620  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3702  015626  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3703  015634  001414                          BEQ     70$             ;BR IF OK
3704  015636  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3705  015644  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3706  015652  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3707  015660  104034                          ERROR   34              ;TYPE MESSAGE 34
3708  015662  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3709  015666  000240                  70$:    NOP
3710  015670  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3711  015676  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3712  015704  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3713  015710  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3714  015716  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3715  015724  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3716  015730  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
```

```
3717  015736  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3718  015744  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3719  015752  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3720  015760  001414                          BEQ     72$             ;BR IF OK
3721  015762  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3722  015770  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3723  015776  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3724  016004  104034                          ERROR   34              ;TYPE MESSAGE 34
3725  016006  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3726  016012  000240                  72$:    NOP
3727
3728                                          ;RELEASE THE DRIVE FROM PORT A
3729
3730  016014  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3731  016022  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3732  016030  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
3733
3734                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3735
3736  016036  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
3737  016042  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3738  016050  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3739  016054  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
3740  016062  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3741  016070  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3742  016076  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3743  016104  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3744  016112  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3745  016120  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3746  016126  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3747  016134  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3748  016142  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3749  016150  001006                          BNE     74$             ;BR IF NOT
3750  016152  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3751  016156  001045                          BNE     76$             ;BR IF NOT
3752  016160  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3753  016162  000137  016362                  JMP     78$             ;BYPASS THE REST OF THE CHECKS
3754  016166  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3755  016174  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3756  016202  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3757  016210  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3758  016214  001414                          BEQ     75$             ;BR IF ZERO
3759  016216  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3760  016224  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3761  016232  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3762  016240  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3763  016244  001012                          BNE     76$             ;BR IF NOT
3764  016246  012737  177777  001250  75$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3765  016254  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
3766  016262  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
3767  016270  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3768  016272  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3769  016300  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3770  016306  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3771  016314  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3772  016322  001401                          BEQ     77$             ;BR IF OK FROM PORT A.
```

```
                                                ERROR   7                       ;REPORT ERROR
                                77$:    MOV     $TMP3,$BDDAT            ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                        MOV     PORTB,PTNBR             ;CHANGE PORT NUMBER
                                        BIC     #ATA,$TMP3              ;DON'T CHECK THE ATTN BIT
                                        CMP     $GDDAT,$TMP3            ;SEE IF READ OK FROM PORT B.
                                        BEQ     78$                     ;BR IF OK
                                        ERROR   7                       ;REPORT ERROR
                                78$:    NOP
                                1$:     SCOPE                           ;LOOP ?

                                ;;*****************************************************************
                                ;*TEST 13       TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
                                ;*
                                ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
                                ;*      PORT TO RELEASE THE DRIVE.
                                ;*
                                ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.
                                ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
                                ;*
                                ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
                                ;*      DOES NOT RETURN TO NEUTRAL.
                                ;*
                                ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
                                ;*      DOES NOT RETURN TO NEUTRAL.
                                ;*
                                ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                                ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                ;*
                                ;;*****************************************************************
                                TST13:
                                        TST     KYBCTL                  ;PERFORMING ONLY SINGLE TESTS ?
                                        BEQ     2$                      ;BR IF NOT
                                        BPL     1$                      ;BR IF JUST ENTERED TEST
                                        JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
                                1$:     MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
                                2$:     MOVB    #13,$TSTNM              ;TEST NUMBER
                                        MOV     #TEST13,$LPADR          ;LOAD LOOP ON TEST ADDRESS
                                        MOV     #TEST13,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
                                        MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS
                                TEST13: MOV     #STACK,SP               ;LOAD THE STACK POINTER

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

                                        ;SEIZE THE DRIVE THROUGH PORT B

                                        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
                                        MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
                                        CLR     RPDS1(R0)               ;WRITE RPDS1
                                        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
                                        MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        MOV     PORTA,OPPRT             ;'OPPOSITE' PORT ADDRESS
                                        MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
                                        MOV     R0,$BDADR               ;RH11 BASE ADDRESS
                                        ADD     #RPDS1,$BDADR           ;GENERATE BAD REGISTER ADDRESS
                                        CLR     $GDDAT                  ;REGISTER SHOULD BE ZERO
                                        CMP     $GDDAT,$BDDAT           ;IS THE REGISTER ZERO
```

```
3829  016540  001403                        BEQ     64$             ;BR IF IT IS
3830  016542  104004                        ERROR   4               ;REPORT THE ERROR
3831  016544  000137  017742                JMP     1$              ;BYPASS REST OF THE SUBTEST
3832  016550                        64$:
3833  016550  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3834  016556  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3835  016564  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
3836  016572  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
3837  016600  013737  001124  001166        MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
3838  016606  005137  001166                COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
3839  016612  013737  001126  001164        MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
3840  016620  043737  001166  001164        BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
3841  016626  023737  001124  001164        CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
3842  016634  001401                        BEQ     65$             ;BR IF THEY ARE
3843  016636  104005                        ERROR 5                 ;REPORT THE ERROR
3844  016640  000240                65$:    NOP
3845
3846                                ;;*****************************************************************
3847                                ;DRIVE CLEAR THROUGH PORT B FIRST
3848
3849  016642  012760  000011  000000        MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT B
3850
3851                                ;;*****************************************************************
3852                                ;VERIFY THAT DRIVE STILL SEIZED BY PORT B
3853
3854  016650  113760  001224  000010        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3855  016656  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3856  016664  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3857  016670  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3858  016676  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3859  016704  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3860  016710  005037  001124                CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3861  016714  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3862  016722  042737  100000  001164        BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3863  016730  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3864  016736  001414                        BEQ     66$             ;BR IF OK
3865  016740  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3866  016746  042737  077777  001174        BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3867  016754  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3868  016762  104033                        ERROR   33              ;TYPE MESSAGE 33
3869  016764  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3870  016770  000240                66$:    NOP
3871  016772  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3872  017000  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3873  017006  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3874  017012  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3875  017020  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3876  017026  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3877  017032  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3878  017040  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3879  017046  042737  100000  001164        BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3880  017054  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3881  017062  001414                        BEQ     68$             ;BR IF OK
3882  017064  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3883  017072  042737  077777  001174        BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3884  017100  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
```

```
3885  017106  104033                          ERROR   33              ;TYPE MESSAGE 33
3886  017110  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3887  017114  000240                  68$:    NOP
3888
3889                                  ;;*****************************************************************
3890                                  ;NOW ISSUE MASSBUS INIT
3891
3892  017116  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;ISSUE MASSBUS INIT
3893
3894                                  ;;*****************************************************************
3895                                  ;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
3896
3897  017124  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
3898  017132  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3899  017140  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3900  017144  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3901  017152  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3902  017160  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3903  017164  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
3904  017170  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3905  017176  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3906  017204  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3907  017212  001414                          BEQ     70$             ;BR IF OK
3908  017214  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3909  017222  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3910  017230  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3911  017236  104034                          ERROR   34              ;TYPE MESSAGE 34
3912  017240  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3913  017244  000240                  70$:    NOP
3914  017246  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3915  017254  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3916  017262  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
3917  017266  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
3918  017274  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3919  017302  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
3920  017306  012737  011700  001124          MOV     #MOL!PGM!DPR!DRV!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
3921  017314  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
3922  017322  042737  100000  001164          BIC     #^C77777,$TMP0  ;SAVE SPECIFIED BITS
3923  017330  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
3924  017336  001414                          BEQ     72$             ;BR IF OK
3925  017340  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
3926  017346  042737  077777  001174          BIC     #77777,$TMP4    ;CLEAR THE MASKED BITS
3927  017354  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
3928  017362  104034                          ERROR   34              ;TYPE MESSAGE 34
3929  017364  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
3930  017370  000240                  72$:    NOP
3931
3932                                  ;RELEASE THE DRIVE FROM PORT B
3933
3934  017372  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
3935  017400  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3936  017406  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
3937
3938                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
3939
3940  017414  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
```

```
3941  017420  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
3942  017426  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
3943  017432  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
3944  017440  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3945  017446  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
3946  017454  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
3947  017462  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3948  017470  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3949  017476  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
3950  017504  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
3951  017512  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
3952  017520  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
3953  017526  001006                          BNE     74$             ;BR IF NOT
3954  017530  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3955  017534  001045                          BNE     76$             ;BR IF NOT
3956  017536  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
3957  017540  000137  017740                  JMP     78$             ;BYPASS THE REST OF THE CHECKS
3958  017544  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3959  017552  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3960  017560  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
3961  017566  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
3962  017572  001414                          BEQ     75$             ;BR IF ZERO
3963  017574  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3964  017602  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
3965  017610  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
3966  017616  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
3967  017622  001012                          BNE     76$             ;BR IF NOT
3968  017624  012737  177777  001250  75$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
3969  017632  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
3970  017640  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
3971  017646  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
3972  017650  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3973  017656  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
3974  017664  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
3975  017672  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
3976  017700  001401                          BEQ     77$             ;BR IF OK FROM PORT A.
3977  017702  104007                          ERROR   7               ;REPORT ERROR
3978  017704  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3979  017712  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
3980  017720  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
3981  017726  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
3982  017734  001401                          BEQ     78$             ;BR IF OK
3983  017736  104007                          ERROR   7               ;REPORT ERROR
3984  017740  000240                  78$:    NOP
3985  017742  000004                  1$:     SCOPE                   ;LOOP ?
3986
3987
3988                                  ;;*****************************************************************
3989                                  ;*TEST 14       TEST RESET ATTENTION 'A' BY MASSBUS CLEAR
3990                                  ;*
3991                                  ;*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
3992                                  ;*      SEIZING PORT.
3993                                  ;*
3994                                  ;*  A.  SET EACH PORT 'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
3995                                  ;*      SET.
3996                                  ;*
```

```
                                        ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                                        ;*
                                        ;*  C.  ISSUE A MASSBUS CLEAR.
                                        ;*
                                        ;*
                                        ;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE ATTENTION
                                        ;*      BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
                                        ;*      'B' IS STILL SET.
                                        ;*
                                        ;;*****************************************************************
017744                                  TST14:
017744  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
017750  001406                                  BEQ     2$              ;BR IF NOT
017752  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
017754  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
017760  012737  177777  001274    1$:           MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
017766  112737  000014  001102    2$:           MOVB    #14,$TSTNM      ;TEST NUMBER
017774  012737  020016  001106                  MOV     #TEST14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
020002  012737  020016  001110                  MOV     #TEST14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
020010  012737  000004  001176                  MOV     #4,$TIMES       ;;DO 4 ITERATIONS
020016  012706  001100                TEST14:   MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                        ;;*****************************************************************

                                                ;SET ATTENTION BITS FOR BOTH PORTS

020022  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT 64$
020030  012760  177777  000014                  MOV     #-1,RPER1(R0)   ;FORCE ERRORS
020036  005060  000014                          CLR     RPER1(R0)       ;CLEAR THE ERRORS
020042  013760  001226  000010                  MOV     PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
020050  005760  000012                64$:      TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
020054  001775                                  BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
020056  012760  177777  000014                  MOV     #-1,RPER1(R0)   ;FORCE ERRORS ON PORT 65$
020064  005060  000014                          CLR     RPER1(R0)       ;CLEAR THE ERRORS
020070  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT "64$" AGAIN
020076  005760  000012                65$:      TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
020102  001775                                  BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT

                                        ;;*****************************************************************
                                        ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

020104  113760  001224  000010                  MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
020112  013737  001224  001234                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020120  005037  001244                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
020124  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
020132  012737  000012  001122                  MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020140  060037  001122                          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
020144  012737  100000  001124                  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
020152  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
020160  042737  077777  001164                  BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
020166  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
020174  001414                                  BEQ     66$             ;BR IF OK
020176  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
020204  042737  100000  001174                  BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
020212  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
020220  104010                                  ERROR   10              ;REPORT THE ERROR
020222  005137  001244                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
```

```
4053  020226  000240                  66$:    NOP
4054  020230  005737  001244                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
4055  020234  001402                          BEQ     .+6             ;BR IF IT WAS
4056  020236  000137  021244                  JMP     i$              ;BYPASS REST OF TEST IF NOT
4057  020242  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4058  020250  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4059  020256  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4060  020262  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4061  020270  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4062  020276  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4063  020302  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4064  020310  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4065  020316  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4066  020324  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4067  020332  001414                          BEQ     68$             ;BR IF OK
4068  020334  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4069  020342  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4070  020350  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4071  020356  104010                          ERROR   10              ;REPORT THE ERROR
4072  020360  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4073  020364  000240                  68$:    NOP
4074  020366  005737  001244                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
4075  020372  001402                          BEQ     .+6             ;BR IF IT WAS
4076  020374  000137  021244                  JMP     i$              ;BYPASS REST OF TEST IF NOT
4077
4078                                  ;;**************************************************************
4079
4080                                          ;SEIZE THE DRIVE THROUGH PORT A
4081
4082  020400  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
4083  020406  013737  001224  001236          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
4084  020414  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
4085  020420  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
4086
4087                                  ;;**************************************************************
4088                                  ;ISSUE MASSBUS INIT TO PORT A
4089
4090  020426  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;MASSBUS INIT
4091  020434  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A AGAIN
4092
4093                                  ;;**************************************************************
4094                                  ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
4095
4096  020442  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4097  020446  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4098  020454  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4099  020462  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4100  020466  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
4101  020472  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4102  020500  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4103  020506  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4104  020514  001414                          BEQ     72$             ;BR IF OK
4105  020516  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4106  020524  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4107  020532  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4108  020540  104047                          ERROR   47              ;TYPE MESSAGE 47
```

```
 4109  020542  005137  001244                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 4110  020546  000240                  72$:      NOP
 4111
 4112                                  ;;*****************************************************************
 4113
 4114                                          ;RELEASE THE DRIVE FROM PORT A
 4115
 4116  020550  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 4117  020556  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4118  020564  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
 4119
 4120                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 4121
 4122  020572  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 4123  020576  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 4124  020604  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 4125  020610  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
 4126  020616  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 4127  020624  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 4128  020632  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 4129  020640  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 4130  020646  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 4131  020654  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 4132  020662  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 4133  020670  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 4134  020676  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 4135  020704  001006                          BNE     74$             ;BR IF NOT
 4136  020706  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 4137  020712  001045                          BNE     76$             ;BR IF NOT
 4138  020714  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 4139  020716  000137  021116                  JMP     78$             ;BYPASS THE REST OF THE CHECKS
 4140  020722  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 4141  020730  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 4142  020736  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 4143  020744  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 4144  020750  001414                          BEQ     75$             ;BR IF ZERO
 4145  020752  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 4146  020760  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 4147  020766  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 4148  020774  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 4149  021000  001012                          BNE     76$             ;BR IF NOT
 4150  021002  012737  177777  001250  75$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 4151  021010  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 4152  021016  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 4153  021024  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 4154  021026  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 4155  021034  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 4156  021042  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
 4157  021050  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 4158  021056  001401                          BEQ     77$             ;BR IF OK FROM PORT A.
 4159  021060  104007                          ERROR   7               ;REPORT ERROR
 4160  021062  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 4161  021070  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 4162  021076  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
 4163  021104  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 4164  021112  001401                          BEQ     78$             ;BR IF OK
```

```
 4165  021114  104007                          ERROR   7               ;REPORT ERROR
 4166  021116  000240                  78$:    NOP
 4167
 4168                                          ;;*****************************************************************
 4169                                          ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)
 4170
 4171  021120  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 4172  021126  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4173  021134  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 4174  021140  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 4175  021146  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 4176  021154  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 4177  021160  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
 4178  021166  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
 4179  021174  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
 4180  021202  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
 4181  021210  001414                          BEQ     79$             ;BR IF OK
 4182  021212  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
 4183  021220  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
 4184  021226  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
 4185  021234  104050                          ERROR   50              ;TYPE MESSAGE 50
 4186  021236  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 4187  021242  000240                  79$:    NOP
 4188  021244  000004                  1$:     SCOPE                   ;LOOP ?
 4189
 4190                                          ;;*****************************************************************
 4191                                          ;*TEST 15       TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
 4192                                          ;*
 4193                                          ;*VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
 4194                                          ;*      SEIZING PORT.
 4195                                          ;*
 4196                                          ;*  A.  SET EACH PORT'S ATTENTION BIT.  VERIFY THAT BOTH ATTENTION BITS
 4197                                          ;*      SET.
 4198                                          ;*
 4199                                          ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 4200                                          ;*
 4201                                          ;*  C.  ISSUE A MASSBUS CLEAR.
 4202                                          ;*
 4203                                          ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE ATTENTION
 4204                                          ;*      BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
 4205                                          ;*      'A' IS STILL SET.
 4206                                          ;*
 4207                                          ;;*****************************************************************
 4208  021246                                  TST15:
 4209  021246  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4210  021252  001406                          BEQ     2$              ;BR IF NOT
 4211  021254  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 4212  021256  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4213  021262  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4214  021270  112737  000015  001102  2$:     MOVB    #15,$TSTNM      ;TEST NUMBER
 4215  021276  012737  021320  001106          MOV     #TEST15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4216  021304  012737  021320  001110          MOV     #TEST15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4217  021312  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 4218  021320  012706  001100          TEST15: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4219                                          ;;*****************************************************************
 4220
```

```
                                        ;SET ATTENTION BITS FOR BOTH PORTS
                                        
021324  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT 64$
021332  012760  177777  000014        MOV     #-1,RPER1(R0)   ;FORCE ERRORS
021340  005060  000014                CLR     RPER1(R0)       ;CLEAR THE ERRORS
021344  013760  001226  000010        MOV     PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
021352  005760  000012        64$:    TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
021356  001775                        BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
021360  012760  177777  000014        MOV     #-1,RPER1(R0)   ;FORCE ERRORS ON PORT 65$
021366  005060  000014                CLR     RPER1(R0)       ;CLEAR THE ERRORS
021372  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT "64$" AGAIN
021400  005760  000012        65$:    TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
021404  001775                        BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT

                                        ;;*****************************************************************
                                        ;CONFIRM THAT BOTH ATTENTION BITS ARE SET

021406  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
021414  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021422  005037  001244                CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
021426  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
021434  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021442  060037  001122                ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
021446  012737  100000  001124        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
021454  013737  001126  001164        MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
021462  042737  077777  001164        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
021470  023737  001124  001164        CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
021476  001414                        BEQ     66$              ;BR IF OK
021500  013737  001126  001174        MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
021506  042737  100000  001174        BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
021514  053737  001174  001124        BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
021522  104010                        ERROR   10               ;REPORT THE ERROR
021524  005137  001244                COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
021530  000240                66$:    NOP
021532  005737  001244                TST     CKERR            ;WAS ATTN BIT FOR PORT B SET ?
021536  001402                        BEQ     .+6              ;BR IF IT WAS
021540  000137  022546                JMP     1$               ;BYPASS REST OF TEST IF NOT
021544  113760  001224  000010        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
021552  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021560  005037  001244                CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
021564  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
021572  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021600  060037  001122                ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
021604  012737  100000  001124        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
021612  013737  001126  001164        MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
021620  042737  077777  001164        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
021626  023737  001124  001164        CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
021634  001414                        BEQ     68$              ;BR IF OK
021636  013737  001126  001174        MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
021644  042737  100000  001174        BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
021652  053737  001174  001124        BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
021660  104010                        ERROR   10               ;REPORT THE ERROR
021662  005137  001244                COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
021666  000240                68$:    NOP
021670  005737  001244                TST     CKERR            ;WAS ATTN BIT FOR PORT A SET ?
021674  001402                        BEQ     .+6              ;BR IF IT WAS
```

```
 4277  021676  000137  022546                  JMP     1$              ;BYPASS REST OF TEST IF NOT
 4278
 4279                                  ;;*****************************************************************
 4280
 4281                                          ;SEIZE THE DRIVE THROUGH PORT B
 4282
 4283  021702  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4284  021710  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 4285  021716  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
 4286  021722  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
 4287
 4288                                  ;;*****************************************************************
 4289                                  ;ISSUE MASSBUS INIT TO PORT B
 4290
 4291  021730  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;MASSBUS INIT
 4292  021736  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B AGAIN
 4293
 4294                                  ;;*****************************************************************
 4295                                  ;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED
 4296
 4297  021744  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 4298  021750  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
 4299  021756  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 4300  021764  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 4301  021770  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 4302  021774  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
 4303  022002  042737  077777  001164          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
 4304  022010  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
 4305  022016  001414                          BEQ     72$             ;BR IF OK
 4306  022020  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
 4307  022026  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
 4308  022034  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
 4309  022042  104047                          ERROR   47              ;TYPE MESSAGE 47
 4310  022044  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 4311  022050  000240                  72$:    NOP
 4312
 4313                                  ;;*****************************************************************
 4314
 4315                                          ;RELEASE THE DRIVE FROM PORT B
 4316
 4317  022052  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 4318  022060  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4319  022066  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
 4320
 4321                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 4322
 4323  022074  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 4324  022100  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 4325  022106  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 4326  022112  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 4327  022120  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 4328  022126  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 4329  022134  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 4330  022142  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 4331  022150  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 4332  022156  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
4333  022164  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4334  022172  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4335  022200  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4336  022206  001006                          BNE     74$             ;BR IF NOT
4337  022210  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4338  022214  001045                          BNE     76$             ;BR IF NOT
4339  022216  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4340  022220  000137  022420                  JMP     78$             ;BYPASS THE REST OF THE CHECKS
4341  022224  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4342  022232  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4343  022240  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4344  022246  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4345  022252  001414                          BEQ     75$             ;BR IF ZERO
4346  022254  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4347  022262  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4348  022270  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4349  022276  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4350  022302  001012                          BNE     76$             ;BR IF NOT
4351  022304  012737  177777  001250  75$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4352  022312  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
4353  022320  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4354  022326  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4355  022330  013737  001170  001126  76$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4356  022336  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4357  022344  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
4358  022352  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4359  022360  001401                          BEQ     77$             ;BR IF OK FROM PORT A.
4360  022362  104007                          ERROR   7               ;REPORT ERROR
4361  022364  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4362  022372  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4363  022400  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
4364  022406  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4365  022414  001401                          BEQ     78$             ;BR IF OK
4366  022416  104007                          ERROR   7               ;REPORT ERROR
4367  022420  000240                  78$:    NOP
4368
4369                                          ;;*****************************************************************
4370                                          ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT A)
4371
4372  022422  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4373  022430  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4374  022436  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4375  022442  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4376  022450  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4377  022456  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4378  022462  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4379  022470  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4380  022476  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4381  022504  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4382  022512  001414                          BEQ     79$             ;BR IF OK
4383  022514  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4384  022522  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4385  022530  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4386  022536  104050                          ERROR   50              ;TYPE MESSAGE 50
4387  022540  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4388  022544  000240                  79$:    NOP
```

```
 4389  022546  000004                  1$:     SCOPE                   ;LOOP ?
 4390
 4391
 4392
 4393                                  ;;*********************************************************************
 4394                                  ;*TEST 16       TEST CLEAR ATTENTION BY MASSBUS INIT - DRIVE IN NEUTRAL
 4395                                  ;*
 4396                                  ;*VERIFY THAT MASSBUS CLEAR DOES NOT RESET ATTENTION BITS WHEN THE
 4397                                  ;*      DRIVE IS IN NEUTRAL.
 4398                                  ;*
 4399                                  ;*  A.  SET THE ATTENTION BITS FOR BOTH PORTS.
 4400                                  ;*
 4401                                  ;*  B.  VERIFY THAT THE DRIVE IS IN NEUTRAL.
 4402                                  ;*
 4403                                  ;*  C.  ISSUE A MASSBUS INIT.  VERIFY THAT NEITHER ATTENTION BIT HAS
 4404                                  ;*      RESET.
 4405                                  ;*
 4406                                  ;;*********************************************************************
 4407  022550                          TST16:
 4408  022550  005737  001274                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4409  022554  001406                         BEQ     2$              ;BR IF NOT
 4410  022556  100002                         BPL     1$              ;BR IF JUST ENTERED TEST
 4411  022560  000137  002602                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4412  022564  012737  177777  001274  1$:    MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4413  022572  112737  000016  001102  2$:    MOVB    #16,$TSTNM      ;TEST NUMBER
 4414  022600  012737  022622  001106         MOV     #TEST16,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4415  022606  012737  022622  001110         MOV     #TEST16,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4416  022614  012737  000004  001176         MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 4417  022622  012706  001100          TEST16: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4418                                  ;;*********************************************************************
 4419
 4420                                          ;SET ATTENTION BITS FOR BOTH PORTS
 4421
 4422  022626  113760  001224  000010         MOVB    PORTA,RPCS2(R0) ;SELECT PORT 64$
 4423  022634  012760  177777  000014         MOV     #-1,RPER1(R0)   ;FORCE ERRORS
 4424  022642  005060  000014                 CLR     RPER1(R0)       ;CLEAR THE ERRORS
 4425  022646  013760  001226  000010         MOV     PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
 4426  022654  005760  000012          64$:   TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
 4427  022660  001775                         BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
 4428  022662  012760  177777  000014         MOV     #-1,RPER1(R0)   ;FORCE ERRORS ON PORT 65$
 4429  022670  005060  000014                 CLR     RPER1(R0)       ;CLEAR THE ERRORS
 4430  022674  113760  001224  000010         MOVB    PORTA,RPCS2(R0) ;SELECT PORT "64$" AGAIN
 4431  022702  005760  000012          65$:   TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
 4432  022706  001775                         BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT
 4433
 4434                                  ;;*********************************************************************
 4435                                  ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
 4436
 4437  022710  113760  001224  000010         MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 4438  022716  013737  001224  001234         MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
 4439  022724  005037  001244                 CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
 4440  022730  016037  000012  001126         MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 4441  022736  012737  000012  001122         MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 4442  022744  060037  001122                 ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
 4443  022750  012737  100000  001124         MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
 4444  022756  013737  001126  001164         MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
```

```
4445  022764  042737  077777  001164          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
4446  022772  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4447  023000  001414                          BEQ     66$             ;BR IF OK
4448  023002  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4449  023010  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4450  023016  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4451  023024  104010                          ERROR   10              ;REPORT THE ERROR
4452  023026  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4453  023032  000240                  66$:    NOP
4454  023034  005737  001244                  TST     CKERR           ;WAS ATTN BIT FOR PORT A SET ?
4455  023040  001402                          BEQ     .+6             ;BR IF IT WAS
4456  023042  000137  024006                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4457  023046  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4458  023054  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4459  023062  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4460  023066  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4461  023074  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4462  023102  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4463  023106  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4464  023114  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4465  023122  042737  077777  001164          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
4466  023130  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4467  023136  001414                          BEQ     68$             ;BR IF OK
4468  023140  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4469  023146  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4470  023154  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4471  023162  104010                          ERROR   10              ;REPORT THE ERROR
4472  023164  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4473  023170  000240                  68$:    NOP
4474  023172  005737  001244                  TST     CKERR           ;WAS ATTN BIT FOR PORT B SET ?
4475  023176  001402                          BEQ     .+6             ;BR IF IT WAS
4476  023200  000137  024006                  JMP     1$              ;BYPASS REST OF TEST IF NOT
4477
4478                          ;;*****************************************************************
4479
4480                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4481
4482  023204  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4483  023210  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4484  023216  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4485  023222  012737  111700  001124          MOV     #111700,$GDDAT  ;COMPARSION CONSTANT
4486  023230  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4487  023236  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4488  023244  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
4489  023252  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4490  023260  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4491  023266  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4492  023274  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4493  023302  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4494  023310  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4495  023316  001006                          BNE     70$             ;BR IF NOT
4496  023320  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4497  023324  001045                          BNE     72$             ;BR IF NOT
4498  023326  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4499  023330  000137  023514                  JMP     74$             ;BYPASS THE REST OF THE CHECKS
4500  023334  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
```

```
4501  023342  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4502  023350  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4503  023356  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4504  023362  001414                          BEQ     71$             ;BR IF ZERO
4505  023364  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4506  023372  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4507  023400  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4508  023406  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4509  023412  001012                          BNE     72$             ;BR IF NOT
4510  023414  012737  177777  001250  71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4511  023422  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
4512  023430  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4513  023436  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4514  023440  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4515  023446  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4516  023454  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4517  023462  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
4518  023464  104007                          ERROR   7               ;REPORT ERROR
4519  023466  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4520  023474  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4521  023502  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4522  023510  001401                          BEQ     74$             ;BR IF OK
4523  023512  104007                          ERROR   7               ;REPORT ERROR
4524  023514  000240                  74$:    NOP
4525  023516  005737  001250                  TST     RELERR          ;WAS DRIVE IN NEUTRAL ?
4526  023522  001402                          BEQ     .+6             ;BR IF IT WAS
4527  023524  000137  024006                  JMP     1$              ;BYPASS RESET OF TEST
4528                                  ;;*****************************************************************
4529                                  ;ISSUE THE MASSBUS INIT
4530
4531  023530  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;ISSUE A MASSBUS INIT
4532
4533                                  ;;*****************************************************************
4534                                  ;CHECK THE ATTENTION BITS OF BOTH PORTS
4535
4536  023536  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4537  023544  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4538  023552  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4539  023556  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4540  023564  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4541  023572  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4542  023576  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4543  023604  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4544  023612  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4545  023620  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4546  023626  001414                          BEQ     75$             ;BR IF OK
4547  023630  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4548  023636  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4549  023644  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4550  023652  104051                          ERROR   51              ;TYPE MESSAGE 51
4551  023654  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4552  023660  000240                  75$:    NOP
4553  023662  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4554  023670  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4555  023676  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4556  023702  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
```

```
4557  023710  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4558  023716  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4559  023722  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4560  023730  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4561  023736  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
4562  023744  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4563  023752  001414                          BEQ     77$             ;BR IF OK
4564  023754  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4565  023762  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4566  023770  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4567  023776  104051                          ERROR   51              ;TYPE MESSAGE 51
4568  024000  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4569  024004  000240                  77$:    NOP
4570  024006  000004                  1$:     SCOPE                   ;LOOP ?
4571
4572
4573                                  ;;*****************************************************************
4574                                  ;*TEST 17       TEST SEIZE BY RPCS1 READ THROUGH PORT 'A'
4575                                  ;*
4576                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
4577                                  ;*
4578                                  ;*  A.  READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'A'; VERIFY THAT
4579                                  ;*      THE DRIVE IS SEIZED.
4580                                  ;*
4581                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE
4582                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4583                                  ;*
4584                                  ;;*****************************************************************
4585  024010                          T$T17:
4586  024010  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
4587  024014  001406                          BEQ     2$              ;BR IF NOT
4588  024016  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
4589  024020  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
4590  024024  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
4591  024032  112737  000017  001102  2$:     MOVB    #17,$TSTNM      ;TEST NUMBER
4592  024040  012737  024062  001106          MOV     #TEST17,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4593  024046  012737  024062  001110          MOV     #TEST17,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4594  024054  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
4595  024062  012706  001100          TEST17: MOV     #STACK,SP       ;LOAD THE STACK POINTER
4596
4597                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
4598
4599  024066  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
4600  024074  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
4601  024100  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
4602  024106  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4603  024114  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
4604  024122  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
4605  024126  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
4606  024134  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4607
4608                                  ;;*****************************************************************
4609
4610                                          ;SEIZE THE DRIVE THROUGH PORT A
4611
4612  024142  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
```

```
4613  024150  013737  001224  001236          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
4614  024156  005760  000000                  TST     RPCS1(R0)     ;READ RHCS1
4615  024162  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4616  024170  013737  001226  001234          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4617  024176  013737  001226  001240          MOV     PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
4618  024204  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;SEE IF DRIVE SEIZED BY PORT A
4619  024212  010037  001122                  MOV     R0,$BDADR     ;RH11 BASE ADDRESS
4620  024216  062737  000012  001122          ADD     #RPDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
4621  024224  005037  001124                  CLR     $GDDAT        ;REGISTER SHOULD BE ZERO
4622  024230  023737  001124  001126          CMP     $GDDAT,$BDDAT ;IS THE REGISTER ZERO
4623  024236  001403                          BEQ     64$           ;BR IF IT IS
4624  024240  104004                          ERROR   4             ;REPORT THE ERROR
4625  024242  000137  024674                  JMP     1$            ;BYPASS REST OF THE SUBTEST
4626  024246                          64$:
4627  024246  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4628  024254  013737  001224  001234          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4629  024262  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
4630  024270  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
4631  024276  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
4632  024304  005137  001166                  COM     $TMP1         ;COMPLEMENT THE EXPECTED STATUS
4633  024310  013737  001126  001164          MOV     $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
4634  024316  043737  001166  001164          BIC     $TMP1,$TMP0   ;CLEAR UNWANTED BITS
4635  024324  023737  001124  001164          CMP     $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
4636  024332  001401                          BEQ     65$           ;BR IF THEY ARE
4637  024334  104005                          ERROR 5               ;REPORT THE ERROR
4638  024336  000240                  65$:    NOP
4639
4640                                  ;;*****************************************************************
4641
4642                                  ;RELEASE THE DRIVE FROM PORT A
4643
4644  024340  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4645  024346  013737  001224  001234          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4646  024354  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
4647
4648                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4649
4650  024362  005037  001250                  CLR     RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4651  024366  012737  000012  001122          MOV     #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4652  024374  060037  001122                  ADD     R0,$BDADR     ;ADD THE I/O BASE ADDRESS
4653  024400  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
4654  024406  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4655  024414  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4656  024422  013737  001170  001164          MOV     $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
4657  024430  042737  100100  001164          BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4658  024436  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4659  024444  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4660  024452  013737  001172  001166          MOV     $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
4661  024460  042737  100100  001166          BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4662  024466  023737  001164  001166          CMP     $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4663  024474  001006                          BNE     66$           ;BR IF NOT
4664  024476  005737  001164                  TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4665  024502  001045                          BNE     68$           ;BR IF NOT
4666  024504  104046                          ERROR   46            ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4667  024506  000137  024672                  JMP     70$           ;BYPASS THE REST OF THE CHECKS
4668  024512  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
```

```
 4669  024520  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 4670  024526  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 4671  024534  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 4672  024540  001414                          BEQ     67$             ;BR IF ZERO
 4673  024542  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 4674  024550  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 4675  024556  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 4676  024564  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 4677  024570  001012                          BNE     68$             ;BR IF NOT
 4678  024572  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 4679  024600  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 4680  024606  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 4681  024614  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 4682  024616  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 4683  024624  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 4684  024632  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 4685  024640  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
 4686  024642  104007                          ERROR   7               ;REPORT ERROR
 4687  024644  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 4688  024652  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 4689  024660  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 4690  024666  001401                          BEQ     70$             ;BR IF OK
 4691  024670  104007                          ERROR   7               ;REPORT ERROR
 4692  024672  000240                  70$:    NOP
 4693  024674  000004                  1$:     SCOPE                   ;LOOP ?
 4694
 4695                                          ;;*****************************************************************
 4696                                          ;*TEST 20       TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
 4697                                          ;*
 4698                                          ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
 4699                                          ;*
 4700                                          ;*  A.  READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'B'; VERIFY THAT
 4701                                          ;*      THE DRIVE IS SEIZED.
 4702                                          ;*
 4703                                          ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
 4704                                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 4705                                          ;*
 4706                                          ;;*****************************************************************
 4707  024676                                  TST20:
 4708  024676  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4709  024702  001406                          BEQ     2$              ;BR IF NOT
 4710  024704  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 4711  024706  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4712  024712  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4713  024720  112737  000020  001102  2$:     MOVB    #20,$TSTNM      ;TEST NUMBER
 4714  024726  012737  024750  001106          MOV     #TEST20,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4715  024734  012737  024750  001110          MOV     #TEST20,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4716  024742  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 4717  024750  012706  001100          TEST20: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4718
 4719                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 4720
 4721  024754  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 4722  024762  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 4723  024766  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 4724  024774  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
```

```
 4725  025002  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 4726  025010  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 4727  025014  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 4728  025022  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 4729
 4730                                  ;;*****************************************************************
 4731
 4732                                          ;SEIZE THE DRIVE THROUGH PORT B
 4733
 4734  025030  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4735  025036  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 4736  025044  005760  000000                  TST     RPCS1(R0)       ;READ RHCS1
 4737  025050  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 4738  025056  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4739  025064  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
 4740  025072  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
 4741  025100  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
 4742  025104  062737  000012  001122          ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
 4743  025112  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
 4744  025116  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
 4745  025124  001403                          BEQ     64$             ;BR IF IT IS
 4746  025126  104004                          ERROR   4               ;REPORT THE ERROR
 4747  025130  000137  025562                  JMP     1$              ;BYPASS REST OF THE SUBTEST
 4748  025134                          64$:
 4749  025134  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4750  025142  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4751  025150  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
 4752  025156  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
 4753  025164  013737  001124  001166          MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
 4754  025172  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
 4755  025176  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
 4756  025204  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
 4757  025212  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
 4758  025220  001401                          BEQ     65$             ;BR IF THEY ARE
 4759  025222  104005                          ERROR 5                 ;REPORT THE ERROR
 4760  025224  000240                  65$:    NOP
 4761
 4762                                  ;;*****************************************************************
 4763
 4764                                          ;RELEASE THE DRIVE FROM PORT B
 4765
 4766  025226  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4767  025234  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4768  025242  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
 4769
 4770                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 4771
 4772  025250  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 4773  025254  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 4774  025262  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 4775  025266  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
 4776  025274  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 4777  025302  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 4778  025310  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 4779  025316  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 4780  025324  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
```

```
4781  025332  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4782  025340  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4783  025346  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4784  025354  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4785  025362  001006                          BNE     66$             ;BR IF NOT
4786  025364  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4787  025370  001045                          BNE     68$             ;BR IF NOT
4788  025372  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4789  025374  000137  025560                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
4790  025400  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4791  025406  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4792  025414  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4793  025422  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4794  025426  001414                          BEQ     67$             ;BR IF ZERO
4795  025430  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4796  025436  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4797  025444  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4798  025452  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4799  025456  001012                          BNE     68$             ;BR IF NOT
4800  025460  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4801  025466  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
4802  025474  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4803  025502  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4804  025504  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4805  025512  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4806  025520  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4807  025526  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
4808  025530  104007                          ERROR   7               ;REPORT ERROR
4809  025532  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4810  025540  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4811  025546  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4812  025554  001401                          BEQ     70$             ;BR IF OK
4813  025556  104007                          ERROR   7               ;REPORT ERROR
4814  025560  000240                  70$:    NOP
4815  025562  000004                  1$:     SCOPE                   ;LOOP ?
4816
4817
4818                                  ;;*****************************************************************
4819                                  ;*TEST 21       TEST 'PORT REQUEST' FROM PORT 'A'
4820                                  ;*
4821                                  ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
4822                                  ;*      DRIVE IS SEIZED BY THE OTHER PORT.
4823                                  ;*
4824                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
4825                                  ;*
4826                                  ;*  B.  WRITE 0'S INTO RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
4827                                  ;*      SEIZED BY PORT 'B'.
4828                                  ;*
4829                                  ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
4830                                  ;*      SWITCHED TO PORT 'A'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
4831                                  ;*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
4832                                  ;*
4833                                  ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
4834                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
4835                                  ;*
4836                                  ;;*****************************************************************
```

```
 4837  025564                                TST21:
 4838  025564  005737  001274                      TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4839  025570  001406                              BEQ     2$              ;BR IF NOT
 4840  025572  100002                              BPL     1$              ;BR IF JUST ENTERED TEST
 4841  025574  000137  002602                      JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4842  025600  012737  177777  001274  1$:         MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4843  025606  112737  000021  001102  2$:         MOVB    #21,$TSTNM      ;TEST NUMBER
 4844  025614  012737  025636  001106              MOV     #TEST21,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4845  025622  012737  025636  001110              MOV     #TEST21,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4846  025630  012737  007640  001176              MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 4847  025636  012706  001100          TEST21:     MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4848
 4849                                              ;CLEAR ATTENTION BITS FOR BOTH PORTS
 4850
 4851  025642  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 4852  025650  005060  000012                      CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 4353  025654  012760  000011  000000              MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 4854  025662  012760  000013  000000              MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 4855  025670  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 4856  025676  005060  000012                      CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 4857  025702  012760  000011  000000              MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 4858  025710  012760  000013  000000              MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 4859
 4860                                          ;;**********************************************************************
 4861
 4862                                              ;SEIZE THE DRIVE THROUGH PORT B
 4863
 4864  025716  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4865  025724  013737  001226  001236              MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 4866  025732  005060  000012                      CLR     RPDS1(R0)       ;WRITE RPDS1
 4867  025736  013737  001224  001240              MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
 4868  025744  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 4869  025752  013737  001224  001234              MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4870
 4871                                          ;;**********************************************************************
 4872                                          ;SET PORT REQUEST
 4873
 4874  025760  005060  000012                      CLR     RPDS1(R0)       ;SET PORT REQUEST FOR PORT A
 4875
 4876                                          ;;**********************************************************************
 4877                                          ;RELEASE THROUGH PORT B.  DRIVE SHOULD SWITCH TO PORT A.
 4878
 4879
 4880                                              ;RELEASE THE DRIVE FROM PORT B
 4881
 4882  025764  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 4883  025772  013737  001226  001234              MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4884  026000  012760  000013  000000              MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
 4885
 4886                                              ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
 4887
 4888  026006  005037  001250                      CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
 4889  026012  012737  111700  001124              MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 4890  026020  012737  000012  001122              MOV     #RPDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
 4891  026026  060037  001122                      ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
 4892  026032  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
```

```
4893  026040  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4894  026046  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
4895  026054  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4896  026062  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4897  026070  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
4898  026076  001404                          BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
4899  026100  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
4900  026104  001401                          BEQ     66$             ;BR IF ZERO
4901  026106  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
4902  026110  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
4903  026116  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
4904  026124  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
4905  026132  001401                          BEQ     67$             ;BR IF OK
4906  026134  104027                          ERROR   27              ;REPORT REGISTER ERROR
4907  026136  000240                  67$:    NOP
4908  026140  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
4909  026146  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4910  026154  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4911  026160  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4912  026166  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4913  026174  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4914  026200  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
4915  026204  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4916  026212  042737  077777  001164          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
4917  026220  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4918  026226  001414                          BEQ     68$             ;BR IF OK
4919  026230  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4920  026236  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4921  026244  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4922  026252  104016                          ERROR   16              ;TYPE MESSAGE 16
4923  026254  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4924  026260  000240                  68$:    NOP
4925  026262  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4926  026270  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
4927  026276  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
4928  026302  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
4929  026310  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
4930  026316  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
4931  026322  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
4932  026330  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
4933  026336  042737  077777  001164          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
4934  026344  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
4935  026352  001414                          BEQ     70$             ;BR IF OK
4936  026354  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
4937  026362  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
4938  026370  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
4939  026376  104016                          ERROR   16              ;TYPE MESSAGE 16
4940  026400  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
4941  026404  000240                  70$:    NOP
4942
4943                                  ;;*****************************************************************
4944
4945                                          ;RELEASE THE DRIVE FROM PORT A
4946
4947  026406  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
4948  026414  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
4949  026422  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
4950
4951                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
4952
4953  026430  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
4954  026434  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
4955  026442  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
4956  026446  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
4957  026454  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4958  026462  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
4959  026470  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
4960  026476  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4961  026504  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4962  026512  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
4963  026520  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
4964  026526  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
4965  026534  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
4966  026542  001006                          BNE     72$             ;BR IF NOT
4967  026544  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
4968  026550  001045                          BNE     74$             ;BR IF NOT
4969  026552  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
4970  026554  000137  026740                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
4971  026560  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
4972  026566  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4973  026574  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
4974  026602  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
4975  026606  001414                          BEQ     73$             ;BR IF ZERO
4976  026610  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
4977  026616  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
4978  026624  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
4979  026632  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
4980  026636  001012                          BNE     74$             ;BR IF NOT
4981  026640  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
4982  026646  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
4983  026654  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
4984  026662  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
4985  026664  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
4986  026672  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
4987  026700  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
4988  026706  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
4989  026710  104007                          ERROR   7               ;REPORT ERROR
4990  026712  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
4991  026720  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
4992  026726  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
4993  026734  001401                          BEQ     76$             ;BR IF OK
4994  026736  104007                          ERROR   7               ;REPORT ERROR
4995  026740  000240                  76$:    NOP
4996  026742  000004                  1$:     SCOPE                   ;LOOP ?
4997
4998                                          ;;*****************************************************************
4999                                          ;*TEST 22       TEST PORT REQUEST FROM PORT 'B'
5000                                          ;*
5001                                          ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
5002                                          ;*      DRIVE IS SEIZED BY THE OTHER PORT.
5003                                          ;*
5004                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
```

```
                                        ;*
                                        ;*  B.  WRITE 0'S INTO RPDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
                                        ;*      SEIZED BY PORT 'A'.
                                        ;*
                                        ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
                                        ;*      SWITCHED TO PORT 'B'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
                                        ;*      PORT 'B' AND IS NOT SET FOR PORT 'A'.
                                        ;*
                                        ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
026744                                  TST22:
026744  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
026750  001406                                  BEQ     2$              ;BR IF NOT
026752  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
026754  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
026760  012737  177777  001274        1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
026766  112737  000022  001102        2$:       MOVB    #22,$TSTNM      ;TEST NUMBER
026774  012737  027016  001106                  MOV     #TEST22,$LPADR  ;LOAD LOOP ON TEST ADDRESS
027002  012737  027016  001110                  MOV     #TEST22,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
027010  012737  007640  001176                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
027016  012706  001100              TEST22:     MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

027022  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
027030  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
027034  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
027042  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
027050  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
027056  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
027062  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
027070  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

                                                ;SEIZE THE DRIVE THROUGH PORT A

027076  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
027104  013737  001224  001236                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
027112  005060  000012                          CLR     RPDS1(R0)       ;WRITE RPDS1
027116  013737  001226  001240                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
027124  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
027132  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                        ;;*****************************************************************
                                        ;SET PORT REQUEST

027140  005060  000012                          CLR     RPDS1(R0)       ;SET PORT REQUEST FOR PORT B

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


                                                ;RELEASE THE DRIVE FROM PORT A
```

```
5061
5062  027144  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5063  027152  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5064  027160  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5065
5066                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5067
5068  027166  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5069  027172  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5070  027200  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5071  027206  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5072  027212  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5073  027220  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5074  027226  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
5075  027234  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5076  027242  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5077  027250  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
5078  027256  001404                          BEQ     66$             ;BR IF STATUS FROM PORT A ZERO
5079  027260  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
5080  027264  001401                          BEQ     66$             ;BR IF ZERO
5081  027266  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
5082  027270  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
5083  027276  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5084  027304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5085  027312  001401                          BEQ     67$             ;BR IF OK
5086  027314  104027                          ERROR   27              ;REPORT REGISTER ERROR
5087  027316  000240                  67$:    NOP
5088  027320  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5089  027326  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5090  027334  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5091  027340  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5092  027346  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5093  027354  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5094  027360  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5095  027364  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5096  027372  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5097  027400  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5098  027406  001414                          BEQ     68$             ;BR IF OK
5099  027410  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5100  027416  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5101  027424  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5102  027432  104016                          ERROR   16              ;TYPE MESSAGE 16
5103  027434  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5104  027440  000240                  68$:    NOP
5105  027442  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5106  027450  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5107  027456  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5108  027462  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5109  027470  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5110  027476  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5111  027502  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
5112  027510  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5113  027516  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5114  027524  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5115  027532  001414                          BEQ     70$             ;BR IF OK
5116  027534  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
```

```
5117  027542  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5118  027550  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5119  027556  104016                          ERROR   16              ;TYPE MESSAGE 16
5120  027560  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5121  027564  000240                  70$:    NOP
5122
5123                                  ;;*****************************************************************
5124
5125                                  ;RELEASE THE DRIVE FROM PORT B
5126
5127  027566  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5128  027574  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5129  027602  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5130
5131                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5132
5133  027610  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5134  027614  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5135  027622  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5136  027626  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5137  027634  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5138  027642  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5139  027650  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5140  027656  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5141  027664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5142  027672  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5143  027700  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5144  027706  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5145  027714  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5146  027722  001006                          BNE     72$             ;BR IF NOT
5147  027724  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5148  027730  001045                          BNE     74$             ;BR IF NOT
5149  027732  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5150  027734  000137  030120                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
5151  027740  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5152  027746  013737  0C1226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5153  027754  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5154  027762  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5155  027766  001414                          BEQ     73$             ;BR IF ZERO
5156  027770  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5157  027776  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5158  030004  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5159  030012  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5160  030016  001012                          BNE     74$             ;BR IF NOT
5161  030020  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5162  030026  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5163  030034  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
5164  030042  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5165  030044  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5166  030052  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5167  030060  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5168  030066  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
5169  030070  104007                          ERROR   7               ;REPORT ERROR
5170  030072  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5171  030100  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5172  030106  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
```

```
                        T22     TEST PORT REQUEST FROM PORT 'B'

030114  001401                  BEQ     76$             ;BR IF OK
030116  104007                  ERROR   7               ;REPORT ERROR
030120  000240          76$:    NOP
030122  000004          1$:     SCOPE                   ;LOOP ?


                        ;;*****************************************************************
                        ;*TEST 23       TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
                        ;*
                        ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
                        ;*      REQUEST'.
                        ;*
                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1.  VERIFY THAT
                        ;*      THE DRIVE HAS BEEN SEIZED.
                        ;*
                        ;*  B.  READ THE CONTROL REGISTER FROM PORT 'A'.  VERIFY THAT 'DVA' IS NOT
                        ;*      SET.
                        ;*
                        ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                        ;*
                        ;;*****************************************************************
030124          TST23:
030124  005737  001274          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
030130  001406                  BEQ     2$              ;BR IF NOT
030132  100002                  BPL     1$              ;BR IF JUST ENTERED TEST
030134  000137  002602          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
030140  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
030146  112737  000023  001102  2$:     MOVB    #23,$TSTNM      ;TEST NUMBER
030154  012737  030176  001106          MOV     #TEST23,$LPADR  ;LOAD LOOP ON TEST ADDRESS
030162  012737  030176  001110          MOV     #TEST23,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
030170  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
030176  012706  001100  TEST23: MOV     #STACK,SP       ;LOAD THE STACK POINTER

                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

030202  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
030210  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
030214  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
030222  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
030230  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
030236  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
030242  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
030250  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

                        ;SEIZE THE DRIVE THROUGH PORT B

030256  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
030264  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
030272  005760  000000                  TST     RPCS1(R0)       ;READ RHCS1
030276  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
030304  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030312  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
030320  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
```

```
5229  030326  010037  001122                    MOV     R0,$BDADR          ;RH11 BASE ADDRESS
5230  030332  062737  000012  001122            ADD     #RPDS1,$BDADR      ;GENERATE BAD REGISTER ADDRESS
5231  030340  005037  001124                    CLR     $GDDAT             ;REGISTER SHOULD BE ZERO
5232  030344  023737  001124  001126            CMP     $GDDAT,$BDDAT      ;IS THE REGISTER ZERO
5233  030352  001403                            BEQ     64$                ;BR IF IT IS
5234  030354  104004                            ERROR   4                  ;REPORT THE ERROR
5235  030356  000137  031132                    JMP     1$                 ;BYPASS REST OF THE SUBTEST
5236  030362                          64$:
5237  030362  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5238  030370  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5239  030376  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
5240  030404  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
5241  030412  013737  001124  001166            MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
5242  030420  005137  001166                    COM     $TMP1              ;COMPLEMENT THE EXPECTED STATUS
5243  030424  013737  001126  001164            MOV     $BDDAT,$TMP0       ;SAVE THE ACTUAL STATUS
5244  030432  043737  001166  001164            BIC     $TMP1,$TMP0        ;CLEAR UNWANTED BITS
5245  030440  023737  001124  001164            CMP     $GDDAT,$TMP0       ;ARE THE EXPECTED STATUS BITS SET ?
5246  030446  001401                            BEQ     65$                ;BR IF THEY ARE
5247  030450  104005                            ERROR 5                    ;REPORT THE ERROR
5248  030452  000240                  65$:      NOP
5249  030454  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5250  030462  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5251
5252                                    ;;*****************************************************************
5253                                    ;READ RPCS1 THROUGH PORT A - TRY TO SET PORT REQUEST
5254
5255  030470  005037  001244                    CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
5256  030474  016037  000000  001126            MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
5257  030502  012737  000000  001122            MOV     #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5258  030510  060037  001122                    ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
5259  030514  005037  001124                    CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
5260  030520  013737  001126  001164            MOV     $BDDAT,$TMP0       ;MOVE REGISTER CONTENTS TO '$TMP0'
5261  030526  042737  173700  001164            BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
5262  030534  023737  001124  001164            CMP     $GDDAT,$TMP0       ;COMPARE THE BITS
5263  030542  001414                            BEQ     66$                ;BR IF OK
5264  030544  013737  001126  001174            MOV     $BDDAT,$TMP4       ;COPY 'BAD DATA'
5265  030552  042737  004077  001174            BIC     #4077,$TMP4        ;CLEAR THE MASKED BITS
5266  030560  053737  001174  001124            BIS     $TMP4,$GDDAT       ;'OR' WITH GOOD DATA FOR TYPEOUT
5267  030566  104010                            ERROR   10                 ;REPORT THE ERROR
5268  030570  005137  001244                    COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
5269  030574  000240                  66$:      NOP
5270
5271                                    ;;*****************************************************************
5272                                    ;DRIVE SHOULD RETURN TO NEUTRAL
5273
5274
5275                                            ;RELEASE THE DRIVE FROM PORT B
5276
5277  030576  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5278  030604  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5279  030612  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5280
5281                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5282
5283  030620  005037  001250                    CLR     RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5284  030624  012737  000012  001122            MOV     #RPDS1,$BDADR      ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
```

```
5285  030632  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5286  030636  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5287  030644  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5288  030652  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5289  030660  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5290  030666  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5291  030674  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5292  030702  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5293  030710  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5294  030716  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5295  030724  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5296  030732  001006                        BNE     68$             ;BR IF NOT
5297  030734  005737  001164                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5298  030740  001045                        BNE     70$             ;BR IF NOT
5299  030742  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5300  030744  000137  031130                JMP     72$             ;BYPASS THE REST OF THE CHECKS
5301  030750  013737  001170  001126  68$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5302  030756  013737  001226  001234        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5303  030764  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5304  030772  005737  001164                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5305  030776  001414                        BEQ     69$             ;BR IF ZERO
5306  031000  013737  001224  001234        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5307  031006  013737  001172  001126        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5308  031014  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5309  031022  005737  001166                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5310  031026  001012                        BNE     70$             ;BR IF NOT
5311  031030  012737  177777  001250  69$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5312  031036  012760  000011  000000        MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5313  031044  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
5314  031052  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
5315  031054  013737  001170  001126  70$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5316  031062  013737  001224  001234        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5317  031070  023737  001124  001170        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5318  031076  001401                        BEQ     71$             ;BR IF OK FROM PORT A.
5319  031100  104007                        ERROR   7               ;REPORT ERROR
5320  031102  013737  001172  001126  71$:  MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5321  031110  013737  001226  001234        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5322  031116  023737  001124  001172        CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5323  031124  001401                        BEQ     72$             ;BR IF OK
5324  031126  104007                        ERROR   7               ;REPORT ERROR
5325  031130  000240                72$:    NOP
5326  031132  000004                1$:     SCOPE                   ;LOOP ?
5327
5328                                ;;*****************************************************************
5329                                ;*TEST 24       TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'B'
5330                                ;*
5331                                ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
5332                                ;*      REQUEST'.
5333                                ;*
5334                                ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY READING RPCS1.  VERIFY THAT
5335                                ;*      THE DRIVE HAS BEEN SEIZED.
5336                                ;*
5337                                ;*  B.  READ THE CONTROL REGISTER FROM PORT 'B'.  VERIFY THAT 'DVA' IS NOT
5338                                ;*      SET.
5339                                ;*
5340                                ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
```

```
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
031134                                  TST24:
031134  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
031140  001406                                  BEQ     2$              ;BR IF NOT
031142  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
031144  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
031150  012737  177777  001274        1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
031156  112737  000024  001102        2$:       MOVB    #24,$TSTNM      ;TEST NUMBER
031164  012737  031206  001106                  MOV     #TEST24,$LPADR  ;LOAD LOOP ON TEST ADDRESS
031172  012737  031206  001110                  MOV     #TEST24,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
031200  012737  007640  001176                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
031206  012706  001100                TEST24:   MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

031212  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
031220  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
031224  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
031232  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
031240  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
031246  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
031252  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
031260  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

                                                ;SEIZE THE DRIVE THROUGH PORT A

031266  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
031274  013737  001224  001236                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
031302  005760  000000                          TST     RPCS1(R0)       ;READ RHCS1
031306  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
031314  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031322  013737  001226  001240                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
031330  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
031336  010037  001122                          MOV     R0,$BDADR       ;RH11 BASE ADDRESS
031342  062737  000012  001122                  ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
031350  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
031354  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
031362  001403                                  BEQ     64$             ;BR IF IT IS
031364  104004                                  ERROR   4               ;REPORT THE ERROR
031366  000137  032142                          JMP     1$              ;BYPASS REST OF THE SUBTEST
031372                                  64$:
031372  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
031400  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
031406  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
031414  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
031422  013737  001124  001166                  MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
031430  005137  001166                          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
031434  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
031442  043737  001166  001164                  BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
031450  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
031456  001401                                  BEQ     65$             ;BR IF THEY ARE
031460  104005                                  ERROR 5                 ;REPORT THE ERROR
```

```
5397  031462  000240                          65$:    NOP
5398  031464  113760  001226  000010                  MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5399  031472  013737  001226  001234                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5400
5401                                          ;;*****************************************************************
5402                                          ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
5403
5404  031500  005037  001244                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5405  031504  016037  000000  001126                  MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
5406  031512  012737  000000  001122                  MOV     #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5407  031520  060037  001122                          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5408  031524  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5409  031530  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5410  031536  042737  173700  001164                  BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
5411  031544  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5412  031552  001414                                  BEQ     66$             ;BR IF OK
5413  031554  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5414  031562  042737  004077  001174                  BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
5415  031570  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5416  031576  104010                                  ERROR   10              ;REPORT THE ERROR
5417  031600  005137  001244                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5418  031604  000240                          66$:    NOP
5419
5420                                          ;;*****************************************************************
5421                                          ;DRIVE SHOULD RETURN TO NEUTRAL.
5422
5423
5424                                                  ;RELEASE THE DRIVE FROM PORT A
5425
5426  031606  113760  001224  000010                  MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5427  031614  013737  001224  001234                  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5428  031622  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5429
5430                                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5431
5432  031630  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5433  031634  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5434  031642  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5435  031646  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5436  031654  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5437  031662  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FRCM PORT A.
5438  031670  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5439  031676  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5440  031704  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5441  031712  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5442  031720  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5443  031726  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5444  031734  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5445  031742  001006                          BNE     68$             ;BR IF NOT
5446  031744  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5447  031750  001045                          BNE     70$             ;BR IF NOT
5448  031752  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5449  031754  000137  032140                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
5450  031760  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5451  031766  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5452  031774  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
```

```
 5453  032002  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 5454  032006  001414                            BEQ     69$             ;BR IF ZERO
 5455  032010  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 5456  032016  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 5457  032024  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 5458  032032  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 5459  032036  001012                            BNE     70$             ;BR IF NOT
 5460  032040  012737  177777  001250  69$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 5461  032046  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 5462  032054  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5463  032062  104026                            ERROR   26              ;TYPE ERROR MESSAGE 26
 5464  032064  013737  001170  001126  70$:     MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 5465  032072  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 5466  032100  023737  001124  001170            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 5467  032106  001401                            BEQ     71$             ;BR IF OK FROM PORT A.
 5468  032110  104007                            ERROR   7               ;REPORT ERROR
 5469  032112  013737  001172  001126  71$:     MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 5470  032120  013737  001226  001234            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 5471  032126  023737  001124  001172            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 5472  032134  001401                            BEQ     72$             ;BR IF OK
 5473  032136  104007                            ERROR   7               ;REPORT ERROR
 5474  032140  000240                    72$:    NOP
 5475  032142  000004                    1$:     SCOPE                   ;LOOP ?
 5476
 5477
 5478                                    ;;*****************************************************************
 5479                                    ;*TEST 25       TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
 5480                                    ;*
 5481                                    ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
 5482                                    ;*      IS SEIZED BY THE OTHER PORT.
 5483                                    ;*
 5484                                    ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 5485                                    ;*
 5486                                    ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
 5487                                    ;*
 5488                                    ;*  C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
 5489                                    ;*
 5490                                    ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE SWITCHED
 5491                                    ;*      TO PORT 'A'.
 5492                                    ;*
 5493                                    ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
 5494                                    ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 5495                                    ;*
 5496                                    ;;*****************************************************************
 5497  032144                            TST25:
 5498  032144  005737  001274                    TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 5499  032150  001406                            BEQ     2$              ;BR IF NOT
 5500  032152  100002                            BPL     1$              ;BR IF JUST ENTERED TEST
 5501  032154  000137  002602                    JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 5502  032160  012737  177777  001274  1$:      MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 5503  032166  112737  000025  001102  2$:      MOVB    #25,$TSTNM      ;TEST NUMBER
 5504  032174  012737  032216  001106            MOV     #TEST25,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 5505  032202  012737  032216  001110            MOV     #TEST25,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 5506  032210  012737  007640  001176            MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 5507  032216  012706  001100          TEST25:   MOV     #STACK,SP       ;LOAD THE STACK POINTER
 5508
```

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

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

                                        ;SEIZE THE DRIVE THROUGH PORT B

                                        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
                                        MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
                                        CLR     RPDS1(R0)       ;WRITE RPDS1
                                        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
                                        MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
                                        MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
                                        MOV     R0,$BDADR       ;RH11 BASE ADDRESS
                                        ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
                                        CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
                                        CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
                                        BEQ     64$             ;BR IF IT IS
                                        ERROR   4               ;REPORT THE ERROR
                                        JMP     1$              ;BYPASS REST OF THE SUBTEST
                                64$:
                                        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
                                        MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
                                        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
                                        MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
                                        COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
                                        MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
                                        BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
                                        CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
                                        BEQ     65$             ;BR IF THEY ARE
                                        ERROR 5                 ;REPORT THE ERROR
                                65$:    NOP

                                ;;**************************************************************
                                ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

                                        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
                                        MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT A

                                ;;**************************************************************
                                ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

                                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
                                        MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
                                        MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
```

```
5565  032536  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5566  032542  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5567  032546  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
5568  032554  001403                          BEQ     66$             ;BR IF OK
5569  032556  104010                          ERROR   10              ;REPORT THE ERROR
5570  032560  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5571  032564  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
5572  032572  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
5573  032600  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
5574  032604  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
5575  032612  001404                          BEQ     67$             ;BR IF NOT
5576  032614  104011                          ERROR   11              ;REPORT THE ERROR
5577  032616  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
5578  032624  000240                  67$:    NOP
5579  032626  005737  001244                  TST     CKERR           ;WAS RPDS1 NON ZERO ?
5580  032632  001402                          BEQ     .+6             ;CONTENTS OF RPDS1 SEEN BY PORT A
5581  032634  000137  033350                  JMP     1$              ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5582
5583                                  ;;**************************************************************
5584
5585                                          ;RELEASE THE DRIVE FROM PORT B
5586
5587  032640  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5588  032646  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5589  032654  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5590
5591                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5592
5593  032662  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5594  032666  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5595  032674  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5596  032702  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5597  032706  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5598  032714  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5599  032722  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
5600  032730  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5601  032736  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5602  032744  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
5603  032752  001404                          BEQ     68$             ;BR IF STATUS FROM PORT B ZERO
5604  032754  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
5605  032760  001401                          BEQ     68$             ;BR IF ZERO
5606  032762  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
5607  032764  013737  001164  001126  68$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
5608  032772  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5609  033000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5610  033006  001401                          BEQ     69$             ;BR IF OK
5611  033010  104027                          ERROR   27              ;REPORT REGISTER ERROR
5612  033012  000240                  69$:    NOP
5613
5614                                          ;RELEASE THE DRIVE FROM PORT A
5615
5616  033014  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5617  033022  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5618  033030  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5619
5620                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
```

```
5621
5622  033036  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5623  033042  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5624  033050  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5625  033054  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5626  033062  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5627  033070  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5628  033076  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5629  033104  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5630  033112  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5631  033120  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5632  033126  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5633  033134  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5634  033142  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5635  033150  001006                          BNE     70$             ;BR IF NOT
5636  033152  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5637  033156  001045                          BNE     72$             ;BR IF NOT
5638  033160  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5639  033162  000137  033346                  JMP     74$             ;BYPASS THE REST OF THE CHECKS
5640  033166  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5641  033174  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5642  033202  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5643  033210  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5644  033214  001414                          BEQ     71$             ;BR IF ZERO
5645  033216  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5646  033224  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5647  033232  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5648  033240  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5649  033244  001012                          BNE     72$             ;BR IF NOT
5650  033246  012737  177777  001250  71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5651  033254  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5652  033262  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
5653  033270  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5654  033272  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5655  033300  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5656  033306  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5657  033314  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
5658  033316  104007                          ERROR   7               ;REPORT ERROR
5659  033320  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5660  033326  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5661  033334  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5662  033342  001401                          BEQ     74$             ;BR IF OK
5663  033344  104007                          ERROR   7               ;REPORT ERROR
5664  033346  000240                  74$:    NOP
5665  033350  000004                  1$:     SCOPE                   ;LOOP ?
5666
5667                                  ;;*****************************************************************
5668                                  ;*TEST 26       TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
5669                                  ;*
5670                                  ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
5671                                  ;*      IS SEIZED BY THE OTHER PORT.
5672                                  ;*
5673                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
5674                                  ;*
5675                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
5676                                  ;*
```

```
                                        ;*  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.
                                        ;*
                                        ;;**********************************************************************
033352                                  TST26:
033352  005737  001274                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
033356  001406                                BEQ     2$              ;BR IF NOT
033360  100002                                BPL     1$              ;BR IF JUST ENTERED TEST
033362  000137  002602                        JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
033366  012737  177777  001274        1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
033374  112737  000026  001102        2$:     MOVB    #26,$TSTNM      ;TEST NUMBER
033402  012737  033424  001106                MOV     #TEST26,$LPADR  ;LOAD LOOP ON TEST ADDRESS
033410  012737  033424  001110                MOV     #TEST26,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
033416  012737  007640  001176                MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
033424  012706  001100            TEST26:     MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS
033430  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
033436  005060  000012                    CLR     RPDS1(R0)       ;SEIZE THE DRIVE
033442  012760  000011  000000            MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
033450  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
033456  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
033464  005060  000012                    CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
033470  012760  000011  000000            MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
033476  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

033504  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
033512  013737  001224  001236            MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
033520  005060  000012                    CLR     RPDS1(R0)       ;WRITE RPDS1
033524  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
033532  013737  001226  001234            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033540  013737  001226  001240            MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
033546  016037  000012  001126            MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
033554  010037  001122                    MOV     R0,$BDADR       ;RH11 BASE ADDRESS
033560  062737  000012  001122            ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
033566  005037  001124                    CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
033572  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
033600  001403                            BEQ     64$             ;BR IF IT IS
033602  104004                            ERROR   4               ;REPORT THE ERROR
033604  000137  034556                    JMP     1$              ;BYPASS REST OF THE SUBTEST
033610                                  64$:
033610  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
033616  013737  001224  001234            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033624  016037  000012  001126            MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
033632  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
033640  013737  001124  001166            MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
```

```
5733  033646  005137  001166                    COM     $TMP1                   ;COMPLEMENT THE EXPECTED STATUS
5734  033652  013737  001126  001164            MOV     $BDDAT,$TMP0            ;SAVE THE ACTUAL STATUS
5735  033660  043737  001166  001164            BIC     $TMP1,$TMP0             ;CLEAR UNWANTED BITS
5736  033666  023737  001124  001164            CMP     $GDDAT,$TMP0            ;ARE THE EXPECTED STATUS BITS SET ?
5737  033674  001401                            BEQ     65$                     ;BR IF THEY ARE
5738  033676  104005                            ERROR 5                         ;REPORT THE ERROR
5739  033700  000240                    65$:    NOP
5740
5741                                            ;;*****************************************************************
5742                                            ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
5743
5744  033702  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5745  033710  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5746  033716  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND THROUGH PORT B
5747
5748                                            ;;*****************************************************************
5749                                            ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
5750
5751  033724  005037  001244                    CLR     CKERR                   ;CLEAR THE 'CHECK ERROR' INDICATOR
5752  033730  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5753  033736  012737  000012  001122            MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5754  033744  060037  001122                    ADD     R0,$BDADR               ;ADD RH11 BASE ADDRESS
5755  033750  005037  001124                    CLR     $GDDAT                  ;WHAT REGISTER SHOULD BE
5756  033754  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;IS THE REGISTER OK ?
5757  033762  001403                            BEQ     66$                     ;BR IF OK
5758  033764  104010                            ERROR   10                      ;REPORT THE ERROR
5759  033766  005137  001244                    COM     CKERR                   ;SET THE REGISTER COMPARE ERROR INDICATOR
5760  033772  016037  000000  001126    66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
5761  034000  012737  000000  001122            MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
5762  034006  060037  001122                    ADD     R0,$BDADR               ;ADDRESS BASE
5763  034012  032737  020000  001126            BIT     #MCPE,$BDDAT            ;IS 'MCPE' SET ?
5764  034020  001404                            BEQ     67$                     ;BR IF NOT
5765  034022  104011                            ERROR   11                      ;REPORT THE ERROR
5766  034024  012760  040000  000000            MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
5767  034032  000240                    67$:    NOP
5768  034034  005737  001244                    TST     CKERR                   ;WAS RPDS1 NON ZERO ?
5769  034040  001402                            BEQ     .+6                     ;CONTENTS OF RPDS1 SEEN BY PORT B
5770  034042  000137  034556                    JMP     1$                      ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5771
5772                                            ;;*****************************************************************
5773
5774                                            ;RELEASE THE DRIVE FROM PORT A
5775
5776  034046  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5777  034054  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5778  034062  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5779
5780                                            ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5781
5782  034070  005037  001250                    CLR     RELERR                  ;CLEAR 'RELEASE ERROR' INDICATOR
5783  034074  012737  111700  001124            MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5784  034102  012737  000012  001122            MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5785  034110  060037  001122                    ADD     R0,$BDADR               ;REGISTER BASE ADDRESS FOR TYPEOUT
5786  034114  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5787  034122  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5788  034130  016037  000012  001164            MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
```

```
5789  034136  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5790  034144  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5791  034152  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
5792  034160  001404                          BEQ     68$             ;BR IF STATUS FROM PORT A ZERO
5793  034162  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
5794  034166  001401                          BEQ     68$             ;BR IF ZERO
5795  034170  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
5796  034172  013737  001164  001126  68$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
5797  034200  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5798  034206  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5799  034214  001401                          BEQ     69$             ;BR IF OK
5800  034216  104027                          ERROR   27              ;REPORT REGISTER ERROR
5801  034220  000240                  69$:    NOP
5802
5803                                          ;RELEASE THE DRIVE FROM PORT B
5804
5805  034222  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5806  034230  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5807  034236  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5808
5809                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5810
5811  034244  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5812  034250  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5813  034256  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5814  034262  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5815  034270  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5816  034276  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5817  034304  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5818  034312  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5819  034320  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5820  034326  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5821  034334  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5822  034342  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5823  034350  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5824  034356  001006                          BNE     70$             ;BR IF NOT
5825  034360  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5826  034364  001045                          BNE     72$             ;BR IF NOT
5827  034366  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5828  034370  000137  034554                  JMP     74$             ;BYPASS THE REST OF THE CHECKS
5829  034374  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5830  034402  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5831  034410  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5832  034416  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5833  034422  001414                          BEQ     71$             ;BR IF ZERO
5834  034424  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5835  034432  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5836  034440  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5837  034446  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5838  034452  001012                          BNE     72$             ;BR IF NOT
5839  034454  012737  177777  001250  71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5840  034462  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5841  034470  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
5842  034476  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
5843  034500  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5844  034506  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
```

```
 5845  034514  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 5846  034522  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
 5847  034524  104007                          ERROR   7               ;REPORT ERROR
 5848  034526  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 5849  034534  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 5850  034542  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 5851  034550  001401                          BEQ     74$             ;BR IF OK
 5852  034552  104007                          ERROR   7               ;REPORT ERROR
 5853  034554  000240                  74$:    NOP
 5854  034556  000004                  1$:     SCOPE                   ;LOOP ?
 5855
 5856
 5857                                  ;;*****************************************************************
 5858                                  ;*TEST 27       TEST SEIZE BY WRITING ATTENTION BIT
 5859                                  ;*
 5860                                  ;*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
 5861                                  ;*      (RPAS) SEIZES THE DRIVE.  VERIFY THAT REQUEST IS SET FOR THE OTHER
 5862                                  ;*      PORT.
 5863                                  ;*
 5864                                  ;*  A.  WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; VERIFY THAT THE DRIVE
 5865                                  ;*      IS SEIZED.
 5866                                  ;*
 5867                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
 5868                                  ;*      DRIVE SWITCHES TO THE OPPOSITE PORT.  ISSUE A RELEASE THROUGH THE
 5869                                  ;*      OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
 5870                                  ;*
 5871                                  ;;*****************************************************************
 5872  034560                          TST27:
 5873  034560  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 5874  034564  001406                          BEQ     2$              ;BR IF NOT
 5875  034566  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 5876  034570  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 5877  034574  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 5878  034602  112737  000027  001102  2$:     MOVB    #27,$TSTNM      ;TEST NUMBER
 5879  034610  012737  034632  001106          MOV     #TEST27,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 5880  034616  012737  034632  001110          MOV     #TEST27,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 5881  034624  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 5882  034632  012706  001100          TEST27: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 5883
 5884                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 5885
 5886  034636  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 5887  034644  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 5888  034650  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 5889  034656  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5890  034664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 5891  034672  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 5892  034676  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 5893  034704  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5894
 5895                                  ;;*****************************************************************
 5896                                  ;SELECT DRIVE OTHER THAN THAT BEING TESTED
 5897
 5898  034712  113760  001230  000010          MOVB    PORTC,RPCS2(R0) ;SELECT DRIVE NOT BEING TESTED
 5899  034720  013737  001224  001236          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
 5900
```

```
5901                                    ;;*****************************************************************
5902                                    ;WRITE THE DRIVE'S ATTENTION BIT
5903
5904   034726  013760  001232  000016           MOV     ASR1,RPAS(R0)   ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
5905   034734  113760  001224  000010           MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5906   034742  013737  001224  001234           MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5907
5908                                    ;;*****************************************************************
5909                                    ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
5910
5911   034750  005760  000012                   TST     RPDS1(R0)       ;SEE THE REGISTER THROUGH PORT A ?
5912   034754  001014                           BNE     1$              ;BR IF YES
5913   034756  113760  001226  000010           MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5914   034764  013737  001226  001234           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5915   034772  005760  000012                   TST     RPDS1(R0)       ;SEE REGISTER THROUGH PORT B ?
5916   034776  001021                           BNE     2$              ;BR IF YES
5917   035000  104042                           ERROR   42              ;DRIVE NOT IN NEUTRAL OR SEIZED
5918   035002  000137  036552                   JMP     4$              ;BYPASS REST OF TEST
5919   035006                          1$:
5920   035006  113760  001226  000010           MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5921   035014  013737  001226  001234           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5922   035022  005760  000012                   TST     RPDS1(R0)       ;REGISTER SHOULD BE ZERO THROUGH PORT B
5923   035026  001002                           BNE     .+6             ;BR IF STATUS REG IS NOT ZERO
5924   035030  000137  035702                   JMP     3$              ;STATUS REG IS ZERO
5925   035034  104043                           ERROR   43              ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
5926   035036  000137  036552                   JMP     4$              ;BYPASS REST OF TEST
5927
5928                                    ;;*****************************************************************
5929                                    ;PORT B HAS THE DRIVE.  VERIFY THAT PORT A HAS PORT REQUEST SET
5930
5931   035042                          2$:
5932   035042  005037  001244                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5933   035046  016037  000012  001126           MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5934   035054  012737  000012  001122           MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5935   035062  060037  001122                   ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5936   035066  012737  011700  001124           MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
5937   035074  013737  001126  001164           MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5938   035102  042737  106077  001164           BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
5939   035110  023737  001124  001164           CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5940   035116  001414                           BEQ     64$             ;BR IF OK
5941   035120  013737  001126  001174           MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5942   035126  042737  071700  001174           BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
5943   035134  053737  001174  001124           BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5944   035142  104010                           ERROR   10              ;REPORT THE ERROR
5945   035144  005137  001244                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5946   035150  000240                  64$:     NOP
5947   035152  013737  001226  001236           MOV     PORTB,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
5948   035160  013737  001224  001240           MOV     PORTA,OPPRT     ;SAME AS ABOVE
5949
5950                                    ;RELEASE THE DRIVE FROM PORT B
5951
5952   035166  113760  001226  000010           MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5953   035174  013737  001226  001234           MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5954   035202  012760  000013  000000           MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5955
5956                                    ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
```

```
5957
5958  035210  005037  001250                    CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5959  035214  012737  111700  001124            MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5960  035222  012737  000012  001122            MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5961  035230  060037  001122                    ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5962  035234  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5963  035242  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5964  035250  016037  000012  001164            MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
5965  035256  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5966  035264  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5967  035272  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
5968  035300  001404                            BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
5969  035302  005737  001164                    TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
5970  035306  001401                            BEQ     66$             ;BR IF ZERO
5971  035310  104044                            ERROR   44              ;REPORT DRIVE NOT SEIZED BY PORT A
5972  035312  013737  001164  001126  66$:     MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
5973  035320  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5974  035326  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5975  035334  001401                            BEQ     67$             ;BR IF OK
5976  035336  104027                            ERROR   27              ;REPORT REGISTER ERROR
5977  035340  000240                    67$:    NOP
5978
5979                                            ;RELEASE THE DRIVE FROM PORT A
5980
5981  035342  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5982  035350  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5983  035356  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5984
5985                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5986
5987  035364  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5988  035370  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5989  035376  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5990  035402  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5991  035410  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5992  035416  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5993  035424  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5994  035432  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5995  035440  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5996  035446  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5997  035454  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5998  035462  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5999  035470  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6000  035476  001006                            BNE     68$             ;BR IF NOT
6001  035500  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6002  035504  001045                            BNE     70$             ;BR IF NOT
6003  035506  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6004  035510  000137  035674                    JMP     72$             ;BYPASS THE REST OF THE CHECKS
6005  035514  013737  001170  001126  68$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6006  035522  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6007  035530  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6008  035536  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6009  035542  001414                            BEQ     69$             ;BR IF ZERO
6010  035544  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6011  035552  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6012  035560  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
```

```
 6013  035566  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 6014  035572  001012                          BNE     70$             ;BR IF NOT
 6015  035574  012737  177777  001250  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 6016  035602  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 6017  035610  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 6018  035616  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 6019  035620  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 6020  035626  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 6021  035634  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 6022  035642  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
 6023  035644  104007                          ERROR   7               ;REPORT ERROR
 6024  035646  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 6025  035654  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 6026  035662  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 6027  035670  001401                          BEQ     72$             ;BR IF OK
 6028  035672  104007                          ERROR   7               ;REPORT ERROR
 6029  035674  000240                  72$:    NOP
 6030  035676  000137  036552                  JMP     4$
 6031
 6032                                          ;;**************************************************************
 6033                                          ;THE DRIVE IS SEIZED BY PORT A.  VERIFY THAT PORT B HAS PORT REQUEST SET
 6034
 6035  035702                          3$:
 6036  035702  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 6037  035710  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6038  035716  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 6039  035722  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 6040  035730  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 6041  035736  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 6042  035742  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
 6043  035750  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
 6044  035756  042737  106077  001164          BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
 6045  035764  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
 6046  035772  001414                          BEQ     73$             ;BR IF OK
 6047  035774  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
 6048  036002  042737  071700  001174          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
 6049  036010  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
 6050  036016  104010                          ERROR   10              ;REPORT THE ERROR
 6051  036020  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 6052  036024  000240                  73$:    NOP
 6053  036026  013737  001224  001236          MOV     PORTA,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
 6054  036034  013737  001226  001240          MOV     PORTB,OPPRT     ;SAME AS ABOVE
 6055
 6056                                          ;RELEASE THE DRIVE FROM PORT A
 6057
 6058  036042  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 6059  036050  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6060  036056  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
 6061
 6062                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
 6063
 6064  036064  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
 6065  036070  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 6066  036076  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
 6067  036104  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
 6068  036110  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
```

```
6069  036116  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6070  036124  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
6071  036132  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6072  036140  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6073  036146  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
6074  036154  001404                          BEQ     75$             ;BR IF STATUS FROM PORT A ZERO
6075  036156  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
6076  036162  001401                          BEQ     75$             ;BR IF ZERO
6077  036164  104044                          ERROR   44              ;REPORT DRIVE NOT SEIZED BY PORT B
6078  036166  013737  001164  001126  75$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
6079  036174  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
6080  036202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
6081  036210  001401                          BEQ     76$             ;BR IF OK
6082  036212  104027                          ERROR   27              ;REPORT REGISTER ERROR
6083  036214  000240                  76$:    NOP
6084
6085                                          ;RELEASE THE DRIVE FROM PORT B
6086
6087  036216  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6088  036224  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6089  036232  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6090
6091                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6092
6093  036240  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6094  036244  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6095  036252  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6096  036256  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6097  036264  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6098  036272  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6099  036300  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6100  036306  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6101  036314  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6102  036322  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6103  036330  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6104  036336  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6105  036344  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6106  036352  001006                          BNE     77$             ;BR IF NOT
6107  036354  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6108  036360  001045                          BNE     79$             ;BR IF NOT
6109  036362  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6110  036364  000137  036550                  JMP     81$             ;BYPASS THE REST OF THE CHECKS
6111  036370  013737  001170  001126  77$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6112  036376  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6113  036404  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6114  036412  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6115  036416  001414                          BEQ     78$             ;BR IF ZERO
6116  036420  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6117  036426  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6118  036434  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6119  036442  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6120  036446  001012                          BNE     79$             ;BR IF NOT
6121  036450  012737  177777  001250  78$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6122  036456  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6123  036464  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6124  036472  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
```

```
6125  036474  013737  001170  001126  79$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6126  036502  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6127  036510  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6128  036516  001401                          BEQ     80$             ;BR IF OK FROM PORT A.
6129  036520  104007                          ERROR   7               ;REPORT ERROR
6130  036522  013737  001172  001126  80$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6131  036530  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6132  036536  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6133  036544  001401                          BEQ     81$             ;BR IF OK
6134  036546  104007                          ERROR   7               ;REPORT ERROR
6135  036550  000240                  81$:    NOP
6136  036552  000004                  4$:     SCOPE                   ;LOOP ?
6137
6138
6139                                  ;;*****************************************************************
6140                                  ;*TEST 30       TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
6141                                  ;*
6142                                  ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
6143                                  ;*      THE DRIVE'S ATTENTION BIT.
6144                                  ;*
6145                                  ;*  A.  SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
6146                                  ;*      BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
6147                                  ;*
6148                                  ;*  B.  VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
6149                                  ;*
6150                                  ;;*****************************************************************
6151  036554                          TST30:
6152  036554  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6153  036560  001406                          BEQ     2$              ;BR IF NOT
6154  036562  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
6155  036564  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6156  036570  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6157  036576  112737  000030  001102  2$:     MOVB    #30,$TSTNM      ;TEST NUMBER
6158  036604  012737  036626  001106          MOV     #TEST30,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6159  036612  012737  036626  001110          MOV     #TEST30,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6160  036620  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
6161  036626  012706  001100          TEST30: MOV     #STACK,SP       ;LOAD THE STACK POINTER
6162
6163                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
6164
6165  036632  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
6166  036640  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
6167  036644  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6168  036652  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6169  036660  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
6170  036666  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
6171  036672  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6172  036700  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6173  036706  113760  001230  000010          MOVB    PORTC,RPCS2(R0) ;SELECT DRIVE NOT BEING TESTED
6174
6175                                  ;;*****************************************************************
6176                                  ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST
6177
6178  036714  013737  001232  001164          MOV     ASR1,$TMP0      ;STORE ATTN BIT FOR PORT A
6179  036722  005137  001164                  COM     $TMP0           ;COMPLEMENT IT
6180  036726  013760  001164  000016          MOV     $TMP0,RPAS(R0)  ;WRITE THE ATTN REGISTER
```

```
                                        ;;*****************************************************************
                                        ;VERIFY THAT DRIVE REMAINDED IN NEUTRAL


                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

036734  005037  001250                        CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
036740  012737  000012  001122                MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
036746  060037  001122                        ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
036752  012737  011700  001124                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
036760  113760  001224  000010                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
036766  016037  000012  001170                MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
036774  013737  001170  001164                MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
037002  042737  100100  001164                BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037010  113760  001226  000010                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
037016  016037  000012  001172                MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
037024  013737  001172  001166                MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
037032  042737  100100  001166                BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037040  023737  001164  001166                CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
037046  001006                                BNE     64$             ;BR IF NOT
037050  005737  001164                        TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
037054  001045                                BNE     66$             ;BR IF NOT
037056  104046                                ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
037060  000137  037244                        JMP     68$             ;BYPASS THE REST OF THE CHECKS
037064  013737  001170  001126  64$:          MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
037072  013737  001226  001234                MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037100  113760  001226  000010                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
037106  005737  001164                        TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
037112  001414                                BEQ     65$             ;BR IF ZERO
037114  013737  001224  001234                MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037122  013737  001172  001126                MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
037130  113760  001224  000010                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
037136  005737  001166                        TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
037142  001012                                BNE     66$             ;BR IF NOT
037144  012737  177777  001250  65$:          MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
037152  012760  000011  000000                MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
037160  012760  000013  000000                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
037166  104021                                ERROR   21              ;TYPE ERROR MESSAGE 21
037170  013737  001170  001126  66$:          MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
037176  013737  001224  001234                MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
037204  023737  001124  001170                CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
037212  001401                                BEQ     67$             ;BR IF OK FROM PORT A.
037214  104007                                ERROR   7               ;REPORT ERROR
037216  013737  001172  001126  67$:          MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
037224  013737  001226  001234                MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
037232  023737  001124  001172                CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
037240  001401                                BEQ     68$             ;BR IF OK
037242  104007                                ERROR   7               ;REPORT ERROR
037244  000240                        68$:     NOP
037246  000004                                SCOPE                   ;LOOP ?


                                        ;;*****************************************************************
                                        ;*TEST 31       TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
                                        ;*
```

```
                                        ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
                                        ;*
                                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                                        ;*
                                        ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
                                        ;*
                                        ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
                                        ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND IS NOT SET FOR
                                        ;*      PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
                                        ;*
                                        ;;*****************************************************************
037250                                  TST31:
037250  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
037254  001406                                  BEQ     2$              ;BR IF NOT
037256  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
037260  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
037264  012737  177777  001274      1$:         MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
037272  112737  000031  001102      2$:         MOVB    #31,$TSTNM      ;TEST NUMBER
037300  012737  037322  001106                  MOV     #TEST31,$LPADR  ;LOAD LOOP ON TEST ADDRESS
037306  012737  037322  001110                  MOV     #TEST31,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
037314  012737  000004  001176                  MOV     #4,$TIMES       ;;DO 4 ITERATIONS
037322  012706  001100              TEST31:     MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                    ;CLEAR ATTENTION BITS FOR BOTH PORTS
037326  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
037334  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
037340  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
037346  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
037354  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
037362  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
037366  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
037374  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                    ;;*****************************************************************
                                    ;SEIZE THE DRIVE THROUGH PORT A
037402  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
037410  013737  001224  001236                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
037416  005060  000012                          CLR     RPDS1(R0)       ;WRITE RPDS1
037422  013737  001226  001240                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
                                    ;;*****************************************************************
                                    ;FORCE AN ERROR
037430  012760  177777  000014                  MOV     #-1,RPER1(R0)   ;SET ERROR BITS
                                    ;;*****************************************************************
                                    ;START THE TIMER
037436  005037  001252                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
037442  012737  003720  001254                  MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
037450  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
037456  013737  001226  001234                  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                    ;;*****************************************************************
```

```
                                        ;WAIT FOR DRIVE TO TIMEOUT

037464  005760  000012        1$:     TST     RPDS1(R0)       ;WAIT FOR THE DRIVE TO BE RELEASED
037470  001004                        BNE     2$              ;BR IF DRIVE RELEASED
037472  005737  001254                TST     WATCH           ;WATCH AT ZERO ?
037476  001372                        BNE     1$              ;BR IF NOT
037500  104036                        ERROR   36              ;DRIVE NOT RELEASED WITHIN 2 SECONDS
037502                          2$:
037502  113760  001224  000010        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
037510  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

037516  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
037522  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
037530  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037536  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
037542  012737  040000  001124        MOV     #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
037550  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
037556  042737  137777  001164        BIC     #^C40000,$TMP0  ;SAVE SPECIFIED BITS
037564  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
037572  001414                        BEQ     66$             ;BR IF OK
037574  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
037602  042737  040000  001174        BIC     #40000,$TMP4    ;CLEAR THE MASKED BITS
037610  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
037616  104023                        ERROR   23              ;TYPE MESSAGE 23
037620  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
037624  000240                  66$:    NOP

                                        ;;**************************************************************
                                        ;THE ERROR REGISTER SHOULD CONTAIN 1'S

037626  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
037632  016037  000014  001126        MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
037640  012737  000014  001122        MOV     #RPER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037646  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
037652  012737  177777  001124        MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
037660  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
037666  001403                        BEQ     68$             ;BR IF OK
037670  104010                        ERROR   10              ;REPORT THE ERROR
037672  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
037676  000240                  68$:    NOP

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

037700  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
037704  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
037712  012737  000012  001122        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037720  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
037724  012737  100000  001124        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
037732  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
037740  042737  077777  001164        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
037746  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
037754  001414                        BEQ     70$             ;BR IF OK
```

```
6349  037756  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6350  037764  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6351  037772  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6352  040000  104041                          ERROR   41              ;TYPE MESSAGE 41
6353  040002  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6354  040006  000240                  70$:    NOP
6355
6356
6357                                  ;;*****************************************************************
6358
6359                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6360
6361  040010  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6362  040014  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6363  040022  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6364  040026  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
6365  040034  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6366  040042  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6367  040050  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6368  040056  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6369  040064  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6370  040072  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6371  040100  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6372  040106  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6373  040114  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6374  040122  001006                          BNE     72$             ;BR IF NOT
6375  040124  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6376  040130  001045                          BNE     74$             ;BR IF NOT
6377  040132  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6378  040134  000137  040334                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6379  040140  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6380  040146  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6381  040154  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6382  040162  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6383  040166  001414                          BEQ     73$             ;BR IF ZERO
6384  040170  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6385  040176  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6386  040204  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6387  040212  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6388  040216  001012                          BNE     74$             ;BR IF NOT
6389  040220  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6390  040226  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6391  040234  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6392  040242  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6393  040244  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6394  040252  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6395  040260  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
6396  040266  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6397  040274  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
6398  040276  104007                          ERROR   7               ;REPORT ERROR
6399  040300  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6400  040306  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6401  040314  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6402  040322  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6403  040330  001401                          BEQ     76$             ;BR IF OK
6404  040332  104007                          ERROR   7               ;REPORT ERROR
```

```
                                T31     TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE

6405  040334  000240           76$:    NOP
6406
6407                            ;;*****************************************************************
6408                            ;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET
6409
6410  040336  113760  001226  000010      MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6411  040344  013737  001226  001234      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6412  040352  005037  001244              CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6413  040356  016037  000012  001126      MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6414  040364  012737  000012  001122      MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6415  040372  060037  001122              ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6416  040376  005037  001124              CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6417  040402  013737  001126  001164      MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6418  040410  042737  077777  001164      BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
6419  040416  023737  001124  001164      CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6420  040424  001414                      BEQ     77$             ;BR IF OK
6421  040426  013737  001126  001174      MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6422  040434  042737  100000  001174      BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6423  040442  053737  001174  001124      BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6424  040450  104052                      ERROR   52              ;TYPE MESSAGE 52
6425  040452  005137  001244              COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6426  040456  000240           77$:    NOP
6427
6428                            ;CLEAR ATTENTION BIT FOR PORT A
6429
6430  040460  113760  001224  000010      MOVB    PORTA,RPCS2(R0)  ;SELECT PORT #A
6431  040466  005060  000012              CLR     RPDS1(R0)       ;SEIZE THE DRIVE
6432  040472  012760  000011  000000      MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6433  040500  012760  000013  000000      MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6434  040506  000004           3$:     SCOPE                   ;LOOP ?
6435
6436                            ;;*****************************************************************
6437                            ;*TEST 32       TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
6438                            ;*
6439                            ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
6440                            ;*
6441                            ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
6442                            ;*
6443                            ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
6444                            ;*
6445                            ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
6446                            ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
6447                            ;*      PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
6448                            ;*
6449                            ;;*****************************************************************
6450  040510                    TST32:
6451  040510  005737  001274              TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6452  040514  001406                      BEQ     2$              ;BR IF NOT
6453  040516  100002                      BPL     1$              ;BR IF JUST ENTERED TEST
6454  040520  000137  002602              JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6455  040524  012737  177777  001274  1$:  MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6456  040532  112737  000032  001102  2$:  MOVB    #32,$TSTNM      ;TEST NUMBER
6457  040540  012737  040562  001106      MOV     #TEST32,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6458  040546  012737  040562  001110      MOV     #TEST32,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6459  040554  012737  000004  001176      MOV     #4,$TIMES       ;;DO 4 ITERATIONS
6460  040562  012706  001100        TEST32: MOV     #STACK,SP       ;LOAD THE STACK POINTER
```

```
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS
                                        ;
040566  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
040574  005060  000012                CLR     RPDS1(R0)       ;SEIZE THE DRIVE
040600  012760  000011  000000        MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
040606  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
040614  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
040622  005060  000012                CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
040626  012760  000011  000000        MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
040634  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                ;;**************************************************************
                                        ;
                                        ;SEIZE THE DRIVE THROUGH PORT B
                                        ;
040642  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
040650  013737  001226  001236        MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
040656  005060  000012                CLR     RPDS1(R0)       ;WRITE RPDS1
040662  013737  001224  001240        MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
                                        ;
                                ;;**************************************************************
                                ;FORCE AN ERROR
                                        ;
040670  012760  177777  000014        MOV     #-1,RPER1(R0)   ;SET ERROR BITS
                                        ;
                                ;;**************************************************************
                                ;START THE TIMER
                                        ;
040676  005037  001252                CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
040702  012737  003720  001254        MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
040710  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
040716  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        ;
                                ;;**************************************************************
                                ;WAIT FOR DRIVE TO TIMEOUT
                                        ;
040724  005760  000012        1$:     TST     RPDS1(R0)       ;WAIT FOR THE DRIVE TO BE RELEASED
040730  001004                        BNE     2$              ;BR IF DRIVE RELEASED
040732  005737  001254                TST     WATCH           ;WATCH AT ZERO ?
040736  001372                        BNE     1$              ;BR IF NOT
040740  104036                        ERROR   36              ;DRIVE NOT RELEASED WITHIN 2 SECONDS
040742                          2$:
040742  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
040750  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        ;
                                ;;**************************************************************
                                ;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET
                                        ;
040756  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040762  016037  000012  001126        MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
040770  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040776  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
041002  012737  040000  001124        MOV     #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
041010  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
041016  042737  137777  001164        BIC     #^C40000,$TMP0  ;SAVE SPECIFIED BITS
041024  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
```

```
6517  041032  001414                          BEQ     66$                 ;BR IF OK
6518  041034  013737  001126  001174          MOV     $BDDAT,$TMP4        ;COPY 'BAD DATA'
6519  041042  042737  040000  001174          BIC     #40000,$TMP4        ;CLEAR THE MASKED BITS
6520  041050  053737  001174  001124          BIS     $TMP4,$GDDAT        ;'OR' WITH GOOD DATA FOR TYPEOUT
6521  041056  104023                          ERROR   23                  ;TYPE MESSAGE 23
6522  041060  005137  001244                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6523  041064  000240                  66$:    NOP
6524
6525                                          ;;*****************************************************************
6526                                          ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6527
6528  041066  005037  001244                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
6529  041072  016037  000014  001126          MOV     RPER1(R0),$BDDAT    ;GET CONTENTS OF RPER1
6530  041100  012737  000014  001122          MOV     #RPER1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6531  041106  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
6532  041112  012737  177777  001124          MOV     #177777,$GDDAT      ;WHAT REGISTER SHOULD BE
6533  041120  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;IS THE REGISTER OK ?
6534  041126  001403                          BEQ     68$                 ;BR IF OK
6535  041130  104010                          ERROR   10                  ;REPORT THE ERROR
6536  041132  005137  001244                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6537  041136  000240                  68$:    NOP
6538
6539                                          ;;*****************************************************************
6540                                          ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
6541
6542  041140  005037  001244                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
6543  041144  016037  000012  001126          MOV     RPDS1(R0),$BDDAT    ;GET CONTENTS OF RPDS1
6544  041152  012737  000012  001122          MOV     #RPDS1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6545  041160  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
6546  041164  012737  100000  001124          MOV     #ATA,$GDDAT         ;WHAT REGISTER SHOULD BE
6547  041172  013737  001126  001164          MOV     $BDDAT,$TMP0        ;MOVE REGISTER CONTENTS TO '$TMP0'
6548  041200  042737  077777  001164          BIC     #^CATA,$TMP0        ;SAVE SPECIFIED BITS
6549  041206  023737  001124  001164          CMP     $GDDAT,$TMP0        ;COMPARE THE BITS
6550  041214  001414                          BEQ     70$                 ;BR IF OK
6551  041216  013737  001126  001174          MOV     $BDDAT,$TMP4        ;COPY 'BAD DATA'
6552  041224  042737  100000  001174          BIC     #ATA,$TMP4          ;CLEAR THE MASKED BITS
6553  041232  053737  001174  001124          BIS     $TMP4,$GDDAT        ;'OR' WITH GOOD DATA FOR TYPEOUT
6554  041240  104041                          ERROR   41                  ;TYPE MESSAGE 41
6555  041242  005137  001244                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
6556  041246  000240                  70$:    NOP
6557
6558
6559                                          ;;*****************************************************************
6560
6561                                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6562
6563  041250  005037  001250                          CLR     RELERR              ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6564  041254  012737  000012  001122                  MOV     #RPDS1,$BDADR       ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6565  041262  060037  001122                          ADD     R0,$BDADR           ;ADD THE I/O BASE ADDRESS
6566  041266  012737  051700  001124                  MOV     #51700,$GDDAT       ;COMPARSION CONSTANT
6567  041274  113760  001224  000010                  MOVB    PORTA,RPCS2(R0)     ;SELECT PORT A.
6568  041302  016037  000012  001170                  MOV     RPDS1(R0),$TMP2     ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6569  041310  013737  001170  001164                  MOV     $TMP2,$TMP0         ;COPY IT INTO '$TMP0'
6570  041316  042737  100100  001164                  BIC     #ATA!VV,$TMP0       ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6571  041324  113760  001226  000010                  MOVB    PORTB,RPCS2(R0)     ;SELECT PORT B.
6572  041332  016037  000012  001172                  MOV     RPDS1(R0),$TMP3     ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
6573  041340  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6574  041346  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6575  041354  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6576  041362  001006                          BNE     72$             ;BR IF NOT
6577  041364  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6578  041370  001045                          BNE     74$             ;BR IF NOT
6579  041372  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6580  041374  000137  041574                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6581  041400  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6582  041406  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6583  041414  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6584  041422  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6585  041426  001414                          BEQ     73$             ;BR IF ZERO
6586  041430  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6587  041436  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6588  041444  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6589  041452  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6590  041456  001012                          BNE     74$             ;BR IF NOT
6591  041460  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6592  041466  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6593  041474  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6594  041502  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6595  041504  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6596  041512  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6597  041520  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
6598  041526  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6599  041534  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
6600  041536  104007                          ERROR   7               ;REPORT ERROR
6601  041540  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6602  041546  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6603  041554  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6604  041562  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6605  041570  001401                          BEQ     76$             ;BR IF OK
6606  041572  104007                          ERROR   7               ;REPORT ERROR
6607  041574  000240                  76$:    NOP
6608
6609                                          ;;*****************************************************************
6610                                          ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET
6611
6612  041576  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6613  041604  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6614  041612  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6615  041616  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6616  041624  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6617  041632  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6618  041636  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6619  041642  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6620  041650  042737  077777  001164          BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
6621  041656  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6622  041664  001414                          BEQ     77$             ;BR IF OK
6623  041666  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6624  041674  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6625  041702  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6626  041710  104052                          ERROR   52              ;TYPE MESSAGE 52
6627  041712  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6628  041716  000240                  77$:    NOP
```

```
                                        ;CLEAR ATTENTION BIT FOR PORT B

041720  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT #B
041726  005060  000012                CLR     RPDS1(R0)        ;SEIZE THE DRIVE
041732  012760  000011  000000        MOV     #11,RPCS1(R0)    ;ISSUE DRIVE CLEAR
041740  012760  000013  000000        MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
041746  000004                3$:     SCOPE                    ;LOOP ?


                        ;;*****************************************************************
                        ;*TEST 33       TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
                        ;*
                        ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
                        ;*      BITS ARE SET IN THE DRIVE.
                        ;*
                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                        ;*
                        ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
                        ;*
                        ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE 'GO'
                        ;*      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
                        ;*      THAT RPER1 HAS NOT BEEN CLEARED.
                        ;*
                        ;*  D.  CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
                        ;*
                        ;*  E.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                        ;*
                        ;;*****************************************************************
041750
041750  005737  001274                TST33:  TST     KYBCTL           ;PERFORMING ONLY SINGLE TESTS ?
041754  001406                                BEQ     2$               ;BR IF NOT
041756  100002                                BPL     1$               ;BR IF JUST ENTERED TEST
041760  000137  002602                        JMP     EXEC             ;RETURN & GET NEXT TEST NUMBER
041764  012737  177777  001274  1$:     MOV     #-1,KYBCTL       ;SET SINGLE TEST INDICATOR
041772  112737  000033  001102  2$:     MOVB    #33,$TSTNM       ;TEST NUMBER
042000  012737  042022  001106        MOV     #TEST33,$LPADR   ;LOAD LOOP ON TEST ADDRESS
042006  012737  042022  001110        MOV     #TEST33,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
042014  012737  007640  001176        MOV     #4000.,$TIMES    ;;DO 4000. ITERATIONS
042022  012706  001100        TEST33: MOV     #STACK,SP        ;LOAD THE STACK POINTER

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

042026  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
042034  005060  000012                CLR     RPDS1(R0)       ;SEIZE THE DRIVE
042040  012760  000011  000000        MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
042046  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
042054  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
042062  005060  000012                CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
042066  012760  000011  000000        MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
042074  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                        ;;*****************************************************************


                                        ;SEIZE THE DRIVE THROUGH PORT A
```

```
6685  042102  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
6686  042110  013737  001224  001236          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
6687  042116  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
6688  042122  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
6689
6690                                  ;;*****************************************************************
6691                                  ;FORCE AN ERROR
6692
6693  042130  012760  177777  000014          MOV     #-1,RPER1(R0)   ;SET ERROR BITS
6694  042136  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
6695  042144  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6696  042150  016037  000000  001126          MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
6697  042156  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6698  042164  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6699  042170  012737  004012  001124          MOV     #4012,$GDDAT    ;WHAT REGISTER SHOULD BE
6700  042176  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6701  042204  042737  173765  001164          BIC     #^C4012,$TMP0   ;SAVE SPECIFIED BITS
6702  042212  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6703  042220  001414                          BEQ     66$             ;BR IF OK
6704  042222  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6705  042230  042737  004012  001174          BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
6706  042236  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6707  042244  104025                          ERROR   25              ;TYPE MESSAGE 25
6708  042246  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6709  042252  000240                  66$:    NOP
6710  042254  005737  001244                  TST     CKERR           ;DID 'GO' BIT RESET ?
6711  042260  001002                          BNE     .+6             ;BR IF NOT
6712  042262  000137  042322                  JMP     1$              ;'GO' BIT RESET
6713  042266  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;INIT THE RH11
6714  042274  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6715  042302  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6716  042310  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT A
6717  042316  000137  043036                  JMP     2$              ;BYPASS THE REST OF THE TEST
6718
6719                                  ;;*****************************************************************
6720                                  ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
6721
6722  042322                          1$:
6723  042322  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6724  042330  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6725  042336  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6726  042342  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6727  042350  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6728  042356  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6729  042362  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6730  042366  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
6731  042374  001403                          BEQ     68$             ;BR IF OK
6732  042376  104024                          ERROR   24              ;TYPE MESSAGE 24
6733  042400  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6734  042404  000240                  68$:    NOP
6735  042406  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6736  042414  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6737  042422  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6738  042426  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
6739  042434  012737  000014  001122          MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6740  042442  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
```

```
6741  042446  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
6742  042454  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
6743  042462  001403                          BEQ     70$             ;BR IF OK
6744  042464  104010                          ERROR   10              ;REPORT THE ERROR
6745  042466  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6746  042472  000240                  70$:    NOP
6747
6748                                          ;;*****************************************************************
6749                                          ;CLEAR THE ERRORS THROUGH PORT A
6750
6751  042474  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
6752
6753                                          ;;*****************************************************************
6754
6755                                          ;RELEASE THE DRIVE FROM PORT A
6756
6757  042502  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
6758  042510  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6759  042516  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
6760
6761                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6762
6763  042524  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6764  042530  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6765  042536  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6766  042542  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6767  042550  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6768  042556  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6769  042564  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6770  042572  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6771  042600  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6772  042606  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6773  042614  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6774  042622  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6775  042630  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6776  042636  001006                          BNE     72$             ;BR IF NOT
6777  042640  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6778  042644  001045                          BNE     74$             ;BR IF NOT
6779  042646  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6780  042650  000137  043034                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6781  042654  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6782  042662  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6783  042670  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6784  042676  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6785  042702  001414                          BEQ     73$             ;BR IF ZERO
6786  042704  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6787  042712  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6788  042720  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6789  042726  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6790  042732  001012                          BNE     74$             ;BR IF NOT
6791  042734  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6792  042742  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6793  042750  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6794  042756  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6795  042760  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6796  042766  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
```

```
6797  042774  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6798  043002  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
6799  043004  104007                          ERROR   7               ;REPORT ERROR
6800  043006  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6801  043014  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6802  043022  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6803  043030  001401                          BEQ     76$             ;BR IF OK
6804  043032  104007                          ERROR   7               ;REPORT ERROR
6805  043034  000240                  76$:    NOP
6806  043036  000004                  2$:     SCOPE                   ;LOOP ?
6807
6808                                  ;;*****************************************************************
6809                                  ;*TEST 34       TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
6810                                  ;*
6811                                  ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
6812                                  ;*      BITS ARE SET IN THE DRIVE.
6813                                  ;*
6814                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
6815                                  ;*
6816                                  ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
6817                                  ;*
6818                                  ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE 'GO'
6819                                  ;*      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
6820                                  ;*      THAT RPER1 HAS NOT BEEN CLEARED.
6821                                  ;*
6822                                  ;*  D.  CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
6823                                  ;*
6824                                  ;*  E.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
6825                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
6826                                  ;*
6827                                  ;;*****************************************************************
6828  043040
6829  043040  005737  001274          TST4:   TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
6830  043044  001406                          BEQ     2$              ;BR IF NOT
6831  043046  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
6832  043050  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
6833  043054  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
6834  043062  112737  000034  001102  2$:     MOVB    #34,$TSTNM      ;TEST NUMBER
6835  043070  012737  043112  001106          MOV     #TEST34,$LPADR  ;LOAD LOOP ON TEST ADDRESS
6836  043076  012737  043112  001110          MOV     #TEST34,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
6837  043104  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
6838  043112  012706  001100          TEST34: MOV     #STACK,SP       ;LOAD THE STACK POINTER
6839
6840                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
6841
6842  043116  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
6843  043124  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
6844  043130  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6845  043136  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6846  043144  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
6847  043152  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
6848  043156  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6849  043164  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6850                                  ;;*****************************************************************
6851
6852                                          ;SEIZE THE DRIVE THROUGH PORT B
```

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

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

                                                MOV     #-1,RPER1(R0)   ;SET ERROR BITS
                                                MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
                                                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
                                                MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
                                                MOV     #RPCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
                                                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
                                                MOV     #4012,$GDDAT    ;WHAT REGISTER SHOULD BE
                                                MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
                                                BIC     #^C4012,$TMP0   ;SAVE SPECIFIED BITS
                                                CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
                                                BEQ     66$             ;BR IF OK
                                                MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
                                                BIC     #4012,$TMP4     ;CLEAR THE MASKED BITS
                                                BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
                                                ERROR   25              ;TYPE MESSAGE 25
                                                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
                                        66$:    NOP
                                                TST     CKERR           ;DID 'GO' BIT RESET ?
                                                BNE     .+6             ;BR IF NOT
                                                JMP     1$              ;'GO' BIT RESET
                                                MOV     #CLR,RPCS2(R0)  ;INIT THE RH11
                                                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                                MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT B
                                                JMP     2$              ;BYPASS THE REST OF THE TEST

                                        ;;**************************************************************
                                        ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B

                                        1$:
                                                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
                                                MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
                                                MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
                                                MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
                                                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
                                                CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
                                                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
                                                BEQ     68$             ;BR IF OK
                                                ERROR   24              ;TYPE MESSAGE 24
                                                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
                                        68$:    NOP
                                                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                                MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
                                                MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
                                                MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
```

```
6909  043532  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6910  043536  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
6911  043544  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
6912  043552  001403                          BEQ     70$             ;BR IF OK
6913  043554  104010                          ERROR   10              ;REPORT THE ERROR
6914  043556  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6915  043562  000240                  70$:    NOP
6916
6917                                  ;;*****************************************************************
6918                                  ;CLEAR THE ERRORS THROUGH PORT B
6919
6920  043564  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
6921
6922                                  ;;*****************************************************************
6923
6924                                  ;RELEASE THE DRIVE FROM PORT B
6925
6926  043572  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
6927  043600  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6928  043606  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6929
6930                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6931
6932  043614  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6933  043620  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6934  043626  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6935  043632  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6936  043640  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6937  043646  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6938  043654  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6939  043662  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6940  043670  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6941  043676  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6942  043704  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6943  043712  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6944  043720  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6945  043726  001006                          BNE     72$             ;BR IF NOT
6946  043730  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6947  043734  001045                          BNE     74$             ;BR IF NOT
6948  043736  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6949  043740  000137  044124                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6950  043744  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6951  043752  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6952  043760  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6953  043766  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6954  043772  001414                          BEQ     73$             ;BR IF ZERO
6955  043774  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6956  044002  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6957  044010  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6958  044016  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6959  044022  001012                          BNE     74$             ;BR IF NOT
6960  044024  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6961  044032  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6962  044040  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6963  044046  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6964  044050  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
```

```
044056  013737  001224  001234            MOV     PORTA,PTNBR       ;CHANGE PORT NUMBER
044064  023737  001124  001170            CMP     $GDDAT,$TMP2      ;ALL BITS OK ?
044072  001401                            BEQ     75$               ;BR IF OK FROM PORT A.
044074  104007                            ERROR   7                 ;REPORT ERROR
044076  013737  001172  001126    75$:    MOV     $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
044104  013737  001226  001234            MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
044112  023737  001124  001172            CMP     $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
044120  001401                            BEQ     76$               ;BR IF OK
044122  104007                            ERROR   7                 ;REPORT ERROR
044124  000240                    76$:    NOP
044126  000004                    2$:     SCOPE                     ;LOOP ?


                                  ;;*****************************************************************
                                  ;*TEST 35        TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
                                  ;*
                                  ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
                                  ;*
                                  ;*   A.   SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                                  ;*
                                  ;*   B.   WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.
                                  ;*
                                  ;*   C.   VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
                                  ;*        TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
                                  ;*
                                  ;*   D.   VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
                                  ;*        BIT IS SET.
                                  ;*
                                  ;;*****************************************************************
044130                            TST35:
044130  005737  001274                    TST     KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
044134  001406                            BEQ     2$                ;BR IF NOT
044136  100002                            BPL     1$                ;BR IF JUST ENTERED TEST
044140  000137  002602                    JMP     EXEC              ;RETURN & GET NEXT TEST NUMBER
044144  012737  177777  001274    1$:     MOV     #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
044152  112737  000035  001102    2$:     MOVB    #35,$TSTNM        ;TEST NUMBER
044160  012737  044202  001106            MOV     #TEST35,$LPADR    ;LOAD LOOP ON TEST ADDRESS
044166  012737  044202  001110            MOV     #TEST35,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
044174  012737  000004  001176            MOV     #4,$TIMES         ;;DO 4 ITERATIONS
044202  012706  001100            TEST35: MOV     #STACK,SP         ;LOAD THE STACK POINTER

                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS

044206  113760  001224  000010            MOVB    PORTA,RPCS2(R0)   ;SELECT PORT #A
044214  005060  000012                    CLR     RPDS1(R0)         ;SEIZE THE DRIVE
044220  012760  000011  000000            MOV     #11,RPCS1(R0)     ;ISSUE DRIVE CLEAR
044226  012760  000013  000000            MOV     #13,RPCS1(R0)     ;RELEASE THE DRIVE
044234  113760  001226  000010            MOVB    PORTB,RPCS2(R0)   ;SELECT PORT #B
044242  005060  000012                    CLR     RPDS1(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
044246  012760  000011  000000            MOV     #11,RPCS1(R0)     ;ISSUE DRIVE CLEAR
044254  012760  000013  000000            MOV     #13,RPCS1(R0)     ;RELEASE THE DRIVE

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

                                          ;SEIZE THE DRIVE THROUGH PORT A
```

```
                                T35     TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

7021  044262  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7022  044270  013737  001224  001236        MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
7023  044276  005060  000012              CLR     RPDS1(R0)       ;WRITE RPDS1
7024  044302  013737  001226  001240        MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
7025
7026                                  ;;*****************************************************************
7027                                  ;WAIT 500 MS
7028
7029
7030                                  ;;*****************************************************************
7031                                  ;START THE TIMER
7032
7033  044310  005037  001252              CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7034  044314  012737  000764  001254        MOV     #500.,WATCH     ;SET WATCH TO 500 MS
7035  044322  005737  001254        1$:   TST     WATCH           ;WATCH EQUAL TO ZERO
7036  044326  001375                      BNE     1$              ;BR IF NOT
7037
7038                                  ;;*****************************************************************
7039                                  ;START THE TIMER
7040
7041  044330  005037  001252              CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7042  044334  012737  003720  001254        MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
7043
7044                                  ;;*****************************************************************
7045                                  ;RETRIGGER THE TIMEOUT ONE-SHOT
7046
7047  044342  005760  000012              TST     RPDS1(R0)       ;RETRIGGER THE ONE-SHOT
7048  044346  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7049  044354  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7050  044362  005760  000012        2$:   TST     RPDS1(R0)       ;WAIT FOR TIMEOUT
7051  044366  001004                      BNE     3$              ;BR IF TIMEOUT OCCURRED
7052  044370  005737  001254              TST     WATCH           ;WATCH EQUAL TO ZERO ?
7053  044374  001372                      BNE     2$              ;BR IF NOT
7054  044376  104036                      ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7055  044400  013737  001252  001272  3$:   MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7056
7057                                  ;;*****************************************************************
7058
7059                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7060
7061  044406  005037  001250              CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7062  044412  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7063  044420  060037  001122              ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7064  044424  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7065  044432  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7066  044440  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7067  044446  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7068  044454  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7069  044462  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7070  044470  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7071  044476  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7072  044504  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7073  044512  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7074  044520  001006                      BNE     66$             ;BR IF NOT
7075  044522  005737  001164              TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7076  044526  001045                      BNE     68$             ;BR IF NOT
```

```
7077  044530  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7078  044532  000137  044716                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
7079  044536  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7080  044544  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7081  044552  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7082  044560  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7083  044564  001414                          BEQ     67$             ;BR IF ZERO
7084  044566  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7085  044574  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7086  044602  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7087  044610  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7088  044614  001012                          BNE     68$             ;BR IF NOT
7089  044616  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7090  044624  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7091  044632  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7092  044640  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
7093  044642  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7094  044650  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7095  044656  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7096  044664  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
7097  044666  104007                          ERROR   7               ;REPORT ERROR
7098  044670  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7099  044676  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7100  044704  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7101  044712  001401                          BEQ     70$             ;BR IF OK
7102  044714  104007                          ERROR   7               ;REPORT ERROR
7103  044716  000240                  70$:    NOP
7104
7105                                          ;;**************************************************************
7106                                          ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
7107
7108  044720  023737  001272  001260          CMP     TIMES,TIMEAP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7109  044726  003004                          BGT     4$              ;BR IF GREATER
7110  044730  023737  001272  001262          CMP     TIMES,TIMEAM    ;MEASURED TIME LESS THAN -25% TOLERANCE
7111  044736  002001                          BGE     .+4             ;BR IF NOT
7112  044740  104025                  4$:     ERROR   25              ;REPORT THE ERROR
7113  044742  000004                          SCOPE                   ;LOOP ?
7114
7115                                          ;;**************************************************************
7116                                          ;*TEST 36       TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
7117                                          ;*
7118                                          ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
7119                                          ;*
7120                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
7121                                          ;*
7122                                          ;*  B.  WAIT 500 MS AND WRITE 0'B INTO RPDS1 THROUGH PORT 'A'.
7123                                          ;*
7124                                          ;*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
7125                                          ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
7126                                          ;*
7127                                          ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
7128                                          ;*      BIT IS SET.
7129                                          ;*
7130                                          ;;**************************************************************
7131  044744                                  TST36:
7132  044744  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
```

```
7133  044750  001406                          BEQ     2$              ;BR IF NOT
7134  044752  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
7135  044754  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
7136  044760  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
7137  044766  112737  000036  001102  2$:     MOVB    #36,$TSTNM      ;TEST NUMBER
7138  044774  012737  045016  001106          MOV     #TEST36,$LPADR  ;LOAD LOOP ON TEST ADDRESS
7139  045002  012737  045016  001110          MOV     #TEST36,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
7140  045010  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
7141  045016  012706  001100          TEST36: MOV     #STACK,SP       ;LOAD THE STACK POINTER
7142
7143                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7144
7145  045022  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
7146  045030  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
7147  045034  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7148  045042  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7149  045050  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
7150  045056  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7151  045062  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7152  045070  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7153
7154                                  ;;**************************************************************
7155
7156                                          ;SEIZE THE DRIVE THROUGH PORT B
7157
7158  045076  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7159  045104  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
7160  045112  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
7161  045116  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
7162
7163                                  ;;**************************************************************
7164                                  ;WAIT 500 MS
7165
7166
7167                                  ;;**************************************************************
7168                                  ;START THE TIMER
7169
7170  045124  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7171  045130  012737  000764  001254          MOV     #500.,WATCH     ;SET WATCH TO 500 MS
7172  045136  005737  001254          1$:     TST     WATCH           ;WATCH EQUAL TO ZERO
7173  045142  001375                          BNE     1$              ;BR IF NOT
7174
7175                                  ;;**************************************************************
7176                                  ;START THE TIMER
7177
7178  045144  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7179  045150  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
7180
7181                                  ;;**************************************************************
7182                                  ;RETRIGGER THE TIMEOUT ONE-SHOT
7183
7184  045156  005760  000012                  TST     RPDS1(R0)       ;RETRIGGER THE ONE-SHOT
7185  045162  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7186  045170  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7187  045176  005760  000012          2$:     TST     RPDS1(R0)       ;WAIT FOR TIMEOUT
7188  045202  001004                          BNE     3$              ;BR IF TIMEOUT OCCURRED
```

```
7189  045204  005737  001254                    TST     WATCH           ;WATCH EQUAL TO ZERO ?
7190  045210  001372                            BNE     2$              ;BR IF NOT
7191  045212  104036                            ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7192  045214  013737  001252  001272  3$:       MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7193
7194                                    ;;*****************************************************************
7195
7196                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7197
7198  045222  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7199  045226  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7200  045234  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7201  045240  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7202  045246  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7203  045254  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7204  045262  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7205  045270  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7206  045276  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7207  045304  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7208  045312  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7209  045320  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7210  045326  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7211  045334  001006                            BNE     66$             ;BR IF NOT
7212  045336  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7213  045342  001045                            BNE     68$             ;BR IF NOT
7214  045344  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7215  045346  000137  045532                    JMP     70$             ;BYPASS THE REST OF THE CHECKS
7216  045352  013737  001170  001126  66$:       MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7217  045360  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7218  045366  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7219  045374  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7220  045400  001414                            BEQ     67$             ;BR IF ZERO
7221  045402  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7222  045410  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7223  045416  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7224  045424  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7225  045430  001012                            BNE     68$             ;BR IF NOT
7226  045432  012737  177777  001250  67$:       MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7227  045440  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7228  045446  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7229  045454  104022                            ERROR   22              ;TYPE ERROR MESSAGE 22
7230  045456  013737  001170  001126  68$:       MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7231  045464  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7232  045472  023737  001124  001170            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7233  045500  001401                            BEQ     69$             ;BR IF OK FROM PORT A.
7234  045502  104007                            ERROR   7               ;REPORT ERROR
7235  045504  013737  001172  001126  69$:       MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7236  045512  013737  001226  001234            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7237  045520  023737  001124  001172            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7238  045526  001401                            BEQ     70$             ;BR IF OK
7239  045530  104007                            ERROR   7               ;REPORT ERROR
7240  045532  000240                  70$:       NOP
7241
7242                                    ;;*****************************************************************
7243                                    ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
7244
```

```
7245  045534  023737  001272  001266          CMP     TIMES,TIMEBP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
7246  045542  003004                          BGT     4$              ;BR IF GREATER
7247  045544  023737  001272  001270          CMP     TIMES,TIMEBM    ;MEASURED TIME LESS THAN -25% TOLERANCE
7248  045552  002001                          BGE     .+4             ;BR IF NOT
7249  045554  104025                  4$:     ERROR   25              ;REPORT THE ERROR
7250  045556  000004                          SCOPE                   ;LOOP ?
7251
7252
7253
7254                                  ;;*****************************************************************
7255                                  ;*TEST 37       TEST PORT 'A' ATTENTION AFTER A COMMAND
7256                                  ;*
7257                                  ;*
7258                                  ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
7259                                  ;*      COMMAND.
7260                                  ;*
7261                                  ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
7262                                  ;*
7263                                  ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
7264                                  ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
7265                                  ;*      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
7266                                  ;*
7267                                  ;*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
7268                                  ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7269                                  ;*
7270                                  ;;*****************************************************************
7271  045560                          TST37:
7272  045560  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7273  045564  001406                          BEQ     2$              ;BR IF NOT
7274  045566  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
7275  045570  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
7276  045574  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
7277  045602  112737  000037  001102  2$:     MOVB    #37,$TSTNM      ;TEST NUMBER
7278  045610  012737  045632  001106          MOV     #TEST37,$LPADR  ;LOAD LOOP ON TEST ADDRESS
7279  045616  012737  045632  001110          MOV     #TEST37,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
7280  045624  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
7281  045632  012706  001100          TEST37: MOV     #STACK,SP       ;LOAD THE STACK POINTER
7282
7283                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7284
7285  045636  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
7286  045644  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
7287  045650  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7288  045656  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7289  045664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
7290  045672  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7291  045676  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7292  045704  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7293  045712  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7294  045720  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7295  045726  013737  001224  001236          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
7296
7297                                  ;;*****************************************************************
7298                                  ;DO A RECALIBRATE THROUGH PORT A
7299
7300  045734  012760  000007  000000          MOV     #7,RPCS1(R0)    ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
```

```
7301
7302                                    ;;*****************************************************************
7303                                    ;WAIT FOR DRIVE TO FINISH
7304
7305  045742  032760  000200  000012          BIT     #DRY,RPDS1(R0)  ;WAIT FOR DRIVE TO FINISH
7306  045750  001774                          BEQ     .-6             ;BR IF NOT FINISHED
7307
7308                                    ;;*****************************************************************
7309                                    ;CONFIRM THAT ATTENTION IS SET FOR PORT A
7310
7311  045752  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7312  045756  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
7313  045764  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7314  045772  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7315  045776  012737  100000  001124          MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
7316  046004  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
7317  046012  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
7318  046020  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
7319  046026  001414                          BEQ     64$             ;BR IF OK
7320  046030  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
7321  046036  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
7322  046044  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
7323  046052  104032                          ERROR   32              ;TYPE MESSAGE 32
7324  046054  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7325  046060  000240                  64$:    NOP
7326  046062  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7327  046070  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7328
7329                                    ;;*****************************************************************
7330                                    ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
7331
7332  046076  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7333  046102  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
7334  046110  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7335  046116  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7336  046122  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7337  046126  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
7338  046134  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
7339  046142  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
7340  046150  001414                          BEQ     66$             ;BR IF OK
7341  046152  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
7342  046160  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
7343  046166  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
7344  046174  104032                          ERROR   32              ;TYPE MESSAGE 32
7345  046176  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7346  046202  000240                  66$:    NOP
7347
7348                                    ;;*****************************************************************
7349
7350                                    ;RELEASE THE DRIVE FROM PORT A
7351
7352  046204  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7353  046212  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7354  046220  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7355
7356                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
```

```
7357
7358  046226  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7359  046232  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7360  046240  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7361  046244  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7362  046252  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7363  046260  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7364  046266  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7365  046274  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7366  046302  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7367  046310  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7368  046316  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7369  046324  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7370  046332  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7371  046340  001006                            BNE     68$             ;BR IF NOT
7372  046342  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7373  046346  001045                            BNE     70$             ;BR IF NOT
7374  046350  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7375  046352  000137  046536                    JMP     72$             ;BYPASS THE REST OF THE CHECKS
7376  046356  013737  001170  001126  68$:      MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7377  046364  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7378  046372  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7379  046400  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7380  046404  001414                            BEQ     69$             ;BR IF ZERO
7381  046406  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7382  046414  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7383  046422  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7384  046430  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7385  046434  001012                            BNE     70$             ;BR IF NOT
7386  046436  012737  177777  001250  69$:      MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7387  046444  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7388  046452  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7389  046460  104026                            ERROR   26              ;TYPE ERROR MESSAGE 26
7390  046462  013737  001170  001126  70$:      MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7391  046470  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7392  046476  023737  001124  001170            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7393  046504  001401                            BEQ     71$             ;BR IF OK FROM PORT A.
7394  046506  104007                            ERROR   7               ;REPORT ERROR
7395  046510  013737  001172  001126  71$:      MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7396  046516  013737  001226  001234            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7397  046524  023737  001124  001172            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7398  046532  001401                            BEQ     72$             ;BR IF OK
7399  046534  104007                            ERROR   7               ;REPORT ERROR
7400  046536  000240                    72$:    NOP
7401  046540  000004                            SCOPE                   ;LOOP ?
7402
7403                                    ;;*****************************************************************
7404                                    ;*TEST 40       TEST PORT 'B' ATTENTION AFTER A COMMAND
7405                                    ;*
7406                                    ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
7407                                    ;*      COMMAND.
7408                                    ;*
7409                                    ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
7410                                    ;*
7411                                    ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
7412                                    ;*      '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.
                                        ;*
                                        ;**************************************************************
046542                                  TST40:
046542 005737 001274                            TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
046546 001406                                   BEQ     2$              ;BR IF NOT
046550 100002                                   BPL     1$              ;BR IF JUST ENTERED TEST
046552 000137 002602                            JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
046556 012737 177777 001274             1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
046564 112737 000040 001102             2$:     MOVB    #40,$TSTNM      ;TEST NUMBER
046572 012737 046614 001106                     MOV     #TEST40,$LPADR  ;LOAD LOOP ON TEST ADDRESS
046600 012737 046614 001110                     MOV     #TEST40,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
046606 012737 000004 001176                     MOV     #4,$TIMES       ;;DO 4 ITERATIONS
046614 012706 001100                    TEST40: MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

046620 113760 001224 000010                     MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
046626 005060 000012                            CLR     RPDS1(R0)       ;SEIZE THE DRIVE
046632 012760 000011 000000                     MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
046640 012760 000013 000000                     MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
046646 113760 001226 000010                     MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
046654 005060 000012                            CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
046660 012760 000011 000000                     MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
046666 012760 000013 000000                     MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
046674 113760 001226 000010                     MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
046702 013737 001226 001234                     MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046710 013737 001226 001236                     MOV     PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS

                                        ;;**************************************************************
                                        ;DO A RECALIBRATE THROUGH PORT B

046716 012760 000007 000000                     MOV     #7,RPCS1(R0)    ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B

                                        ;;**************************************************************
                                        ;WAIT FOR DRIVE TO FINISH

046724 032760 000200 000012                     BIT     #DRY,RPDS1(R0)  ;WAIT FOR DRIVE TO FINISH
046732 001774                                   BEQ     .-6             ;BR IF NOT FINISHED

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

046734 005037 001244                            CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
046740 016037 000012 001126                     MOV     RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
046746 012737 000012 001122                     MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
046754 060037 001122                            ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
046760 012737 100000 001124                     MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
046766 013737 001126 001164                     MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
046774 042737 077777 001164                     BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
047002 023737 001124 001164                     CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
047010 001414                                   BEQ     64$             ;BR IF OK
047012 013737 001126 001174                     MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
```

```
7469  047020  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
7470  047026  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
7471  047034  104032                          ERROR   32              ;TYPE MESSAGE 32
7472  047036  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7473  047042  000240                  64$:    NOP
7474  047044  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7475  047052  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7476
7477                                  ;;*****************************************************************
7478                                  ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A
7479
7480  047060  005037  001244          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7481  047064  016037  000012  001126  MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
7482  047072  012737  000012  001122  MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7483  047100  060037  001122          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7484  047104  005037  001124          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7485  047110  013737  001126  001164  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
7486  047116  042737  077777  001164  BIC     #↑CATA,$TMP0  ;SAVE SPECIFIED BITS
7487  047124  023737  001124  001164  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
7488  047132  001414                  BEQ     66$             ;BR IF OK
7489  047134  013737  001126  001174  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
7490  047142  042737  100000  001174  BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
7491  047150  053737  001174  001124  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
7492  047156  104032                  ERROR   32              ;TYPE MESSAGE 32
7493  047160  005137  001244          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7494  047164  000240                  66$:    NOP
7495
7496                                  ;;*****************************************************************
7497
7498                                  ;RELEASE THE DRIVE FROM PORT B
7499
7500  047166  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7501  047174  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7502  047202  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
7503
7504                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7505
7506  047210  005037  001250          CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7507  047214  012737  000012  001122  MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7508  047222  060037  001122          ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7509  047226  012737  011700  001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7510  047234  113760  001224  000010  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7511  047242  016037  000012  001170  MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7512  047250  013737  001170  001164  MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7513  047256  042737  100100  001164  BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7514  047264  113760  001226  000010  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7515  047272  016037  000012  001172  MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7516  047300  013737  001172  001166  MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7517  047306  042737  100100  001166  BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7518  047314  023737  001164  001166  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7519  047322  001006                  BNE     68$             ;BR IF NOT
7520  047324  005737  001164          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7521  047330  001045                  BNE     70$             ;BR IF NOT
7522  047332  104046                  ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7523  047334  000137  047520          JMP     72$             ;BYPASS THE REST OF THE CHECKS
7524  047340  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
```

```
7525  047346  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7526  047354  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7527  047362  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7528  047366  001414                          BEQ     69$             ;BR IF ZERO
7529  047370  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7530  047376  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7531  047404  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7532  047412  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7533  047416  001012                          BNE     70$             ;BR IF NOT
7534  047420  012737  177777  001250  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7535  047426  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7536  047434  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7537  047442  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7538  047444  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7539  047452  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7540  047460  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7541  047466  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
7542  047470  104007                          ERROR   7               ;REPORT ERROR
7543  047472  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7544  047500  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7545  047506  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7546  047514  001401                          BEQ     72$             ;BR IF OK
7547  047516  104007                          ERROR   7               ;REPORT ERROR
7548  047520  000240                  72$:    NOP
7549  047522  000004                          SCOPE                   ;LOOP ?
7550
7551
7552                                  ;*****************************************************************
7553                                  ;*TEST 41       TEST PORT INTERACTION FROM PORT 'A'
7554                                  ;*
7555                                  ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
7556                                  ;*
7557                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
7558                                  ;*
7559                                  ;*  B.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
7560                                  ;*
7561                                  ;*  C.  READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'.  VERIFY THAT PORT
7562                                  ;*      'B' SEES 0'S FROM EACH OF THESE REGISTERS.
7563                                  ;*
7564                                  ;*  D.  CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
7565                                  ;*
7566                                  ;*  E.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.  VERIFY THAT
7567                                  ;*      PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
7568                                  ;*
7569                                  ;*  F.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE HAS
7570                                  ;*      SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
7571                                  ;*      SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
7572                                  ;*
7573                                  ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
7574                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7575                                  ;*
7576                                  ;*****************************************************************
7577  047524
7578  047524  005737  001274  TST41:  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7579  047530  001406                  BEQ     2$              ;BR IF NOT
7580  047532  100002                  BPL     1$              ;BR IF JUST ENTERED TEST
```

```
7581  047534  000137  002602                    JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
7582  047540  012737  177777  001274  1$:       MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
7583  047546  112737  000041  001102  2$:       MOVB    #41,$TSTNM      ;TEST NUMBER
7584  047554  012737  047576  001106            MOV     #TEST41,$LPADR  ;LOAD LOOP ON TEST ADDRESS
7585  047562  012737  047576  001110            MOV     #TEST41,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
7586  047570  012737  007640  001176            MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
7587  047576  012706  001100          TEST41:   MOV     #STACK,SP       ;LOAD THE STACK POINTER
7588
7589                                            ;CLEAR ATTENTION BITS FOR BOTH PORTS
7590
7591  047602  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
7592  047610  005060  000012                    CLR     RPDS1(R0)       ;SEIZE THE DRIVE
7593  047614  012760  000011  000000            MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7594  047622  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7595  047630  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
7596  047636  005060  000012                    CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7597  047642  012760  000011  000000            MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7598  047650  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7599
7600                                            ;SEIZE THE DRIVE THROUGH PORT A
7601
7602  047656  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7603  047664  013737  001224  001236            MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
7604  047672  005060  000012                    CLR     RPDS1(R0)       ;WRITE RPDS1
7605  047676  013737  001226  001240            MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
7606  047704  012760  177777  000014            MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT A
7607  047712  012760  177777  000040            MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT A
7608  047720  012760  177777  000042            MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT A
7609  047726  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7610  047734  013737  001226  001234            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7611  047742  004737  050600                    JSR     PC,TST41B       ;CHECK THE REGISTERS THROUGH PORT B
7612  047746  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7613  047754  013737  001224  001234            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7614  047762  005060  000042                    CLR     RPER3(R0)       ;CLEAR RPER3 ON PORT A
7615  047766  005060  000040                    CLR     RPER2(R0)       ;CLEAR RPER2 ON PORT A
7616  047772  005060  000014                    CLR     RPER1(R0)       ;CLEAR RPER1 ON PORT A
7617  047776  013760  001232  000016            MOV     ASR1,RPAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT A
7618  050004  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7619  050012  013737  001226  001234            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7620  050020  012760  177777  000014            MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT B
7621  050026  012760  177777  000040            MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT B
7622  050034  012760  177777  000042            MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT B
7623  050042  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7624  050050  013737  001224  001234            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7625  050056  004737  050600                    JSR     PC,TST41B       ;CHECK THE REGISTERS THROUGH PORT A
7626
7627                                            ;RELEASE THE DRIVE FROM PORT A
7628
7629  050062  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7630  050070  013737  001224  001234            MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7631  050076  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7632
7633                                            ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
7634
7635  050104  005037  001250                    CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
7636  050110  012737  111700  001124            MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
```

```
7637  050116  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
7638  050124  060037  001122                  ADD     R0,$BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
7639  050130  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7640  050136  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7641  050144  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
7642  050152  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7643  050160  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7644  050166  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
7645  050174  001404                          BEQ     66$            ;BR IF STATUS FROM PORT A ZERO
7646  050176  005737  001164                  TST     $TMP0          ;IS STATUS FROM PORT B ZERO ?
7647  050202  001401                          BEQ     66$            ;BR IF ZERO
7648  050204  104031                          ERROR   31             ;REPORT DRIVE IN NEUTRAL
7649  050206  013737  001164  001126  66$:   MOV     $TMP0,$BDDAT   ;CHECK STATUS FROM PORT B
7650  050214  013737  001226  001234          MOV     PORTB,PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
7651  050222  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
7652  050230  001401                          BEQ     67$            ;BR IF OK
7653  050232  104027                          ERROR   27             ;REPORT REGISTER ERROR
7654  050234  000240                  67$:    NOP
7655
7656                                          ;RELEASE THE DRIVE FROM PORT B
7657
7658  050236  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7659  050244  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7660  050252  012760  000013  000000          MOV     #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT B
7661
7662                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7663
7664  050260  005037  001250                  CLR     RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7665  050264  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7666  050272  060037  001122                  ADD     R0,$BDADR      ;ADD THE I/O BASE ADDRESS
7667  050276  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7668  050304  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7669  050312  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7670  050320  013737  001170  001164          MOV     $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
7671  050326  042737  100100  001164          BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7672  050334  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7673  050342  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7674  050350  013737  001172  001166          MOV     $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
7675  050356  042737  100100  001166          BIC     #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7676  050364  023737  001164  001166          CMP     $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7677  050372  001006                          BNE     68$            ;BR IF NOT
7678  050374  005737  001164                  TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7679  050400  001045                          BNE     70$            ;BR IF NOT
7680  050402  104046                          ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7681  050404  000137  050570                  JMP     72$            ;BYPASS THE REST OF THE CHECKS
7682  050410  013737  001170  001126  68$:   MOV     $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7683  050416  013737  001226  001234          MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7684  050424  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7685  050432  005737  001164                  TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
7686  050436  001414                          BEQ     69$            ;BR IF ZERO
7687  050440  013737  001224  001234          MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7688  050446  013737  001172  001126          MOV     $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
7689  050454  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7690  050462  005737  001166                  TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
7691  050466  001012                          BNE     70$            ;BR IF NOT
7692  050470  012737  177777  001250  69$:   MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
```

```
7693  050476  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7694  050504  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7695  050512  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7696  050514  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7697  050522  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7698  050530  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7699  050536  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
7700  050540  104007                          ERROR   7               ;REPORT ERROR
7701  050542  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7702  050550  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7703  050556  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7704  050564  001401                          BEQ     72$             ;BR IF OK
7705  050566  104007                          ERROR   7               ;REPORT ERROR
7706  050570  000240                  72$:    NOP
7707  050572  000004                          SCOPE                   ;LOOP ?
7708  050574  000137  051132                  JMP     TST42           ;GO TO THE NEXT TEST
7709
7710                                          ;CHECK THE REGISTERS ON THE SELECTED PORT
7711
7712  050600                          TST41B:
7713  050600  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7714  050604  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
7715  050612  012737  000014  001122          MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7716  050620  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7717  050624  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7718  050630  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7719  050636  001403                          BEQ     64$             ;BR IF OK
7720  050640  104006                          ERROR   6               ;TYPE MESSAGE 6
7721  050642  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7722  050646  016037  000000  001126  64$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7723  050654  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
7724  050662  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
7725  050666  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
7726  050674  001404                          BEQ     65$             ;BR IF NOT
7727  050676  104011                          ERROR   11              ;REPORT THE ERROR
7728  050700  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
7729  050706  000240                  65$:    NOP
7730  050710  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7731  050714  016037  000040  001126          MOV     RPER2(R0),$BDDAT  ;GET CONTENTS OF RPER2
7732  050722  012737  000040  001122          MOV     #RPER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7733  050730  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7734  050734  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7735  050740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7736  050746  001403                          BEQ     66$             ;BR IF OK
7737  050750  104006                          ERROR   6               ;TYPE MESSAGE 6
7738  050752  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7739  050756  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7740  050764  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
7741  050772  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
7742  050776  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
7743  051004  001404                          BEQ     67$             ;BR IF NOT
7744  051006  104011                          ERROR   11              ;REPORT THE ERROR
7745  051010  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
7746  051016  000240                  67$:    NOP
7747  051020  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7748  051024  016037  000042  001126          MOV     RPER3(R0),$BDDAT  ;GET CONTENTS OF RPER3
```

```
7749  051032  012737  000042  001122          MOV     #RPER3,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7750  051040  060037  001122                  ADD     R0,$BDADR      ;ADD RH11 BASE ADDRESS
7751  051044  005037  001124                  CLR     $GDDAT         ;WHAT REGISTER SHOULD BE
7752  051050  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;IS THE REGISTER OK ?
7753  051056  001403                          BEQ     68$            ;BR IF OK
7754  051060  104006                          ERROR   6              ;TYPE MESSAGE 6
7755  051062  005137  001244                  COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
7756  051066  016037  000000  001126  68$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7757  051074  012737  000000  001122          MOV     #RPCS1,$BDADR  ;FORM ADDRESS OF REGISTER
7758  051102  060037  001122                  ADD     R0,$BDADR      ;ADDRESS BASE
7759  051106  032737  020000  001126          BIT     #MCPE,$BDDAT   ;IS 'MCPE' SET ?
7760  051114  001404                          BEQ     69$            ;BR IF NOT
7761  051116  104011                          ERROR   11             ;REPORT THE ERROR
7762  051120  012760  040000  000000          MOV     #TRE,RPCS1(R0) ;CLEAR 'MCPE'
7763  051126  000240                  69$:    NOP
7764  051130  000207                          RTS     PC             ;RETURN
7765
7766                                  ;;*****************************************************************
7767                                  ;*TEST 42       TEST PORT INTERACTION FROM PORT 'B'
7768                                  ;*
7769                                  ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
7770                                  ;*
7771                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
7772                                  ;*
7773                                  ;*  B.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
7774                                  ;*
7775                                  ;*  C.  READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  VERIFY THAT PORT
7776                                  ;*      'A' SEES 0'S FROM EACH OF THESE REGISTERS.
7777                                  ;*
7778                                  ;*  D.  CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
7779                                  ;*
7780                                  ;*  E.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  VERIFY THAT
7781                                  ;*      PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
7782                                  ;*
7783                                  ;*  F.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE HAS
7784                                  ;*      SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
7785                                  ;*      SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
7786                                  ;*
7787                                  ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
7788                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
7789                                  ;*
7790                                  ;*****************************************************************
7791  051132                          TST42:
7792  051132  005737  001274                  TST     KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
7793  051136  001406                          BEQ     2$             ;BR IF NOT
7794  051140  100002                          BPL     1$             ;BR IF JUST ENTERED TEST
7795  051142  000137  002602                  JMP     EXEC           ;RETURN & GET NEXT TEST NUMBER
7796  051146  012737  177777  001274  1$:     MOV     #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
7797  051154  112737  000042  001102  2$:     MOVB    #42,$TSTNM     ;TEST NUMBER
7798  051162  012737  051204  001106          MOV     #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
7799  051170  012737  051204  001110          MOV     #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
7800  051176  012737  007640  001176          MOV     #4000.,$TIMES  ;;DO 4000. ITERATIONS
7801  051204  012706  001100          TEST42: MOV     #STACK,SP      ;LOAD THE STACK POINTER
7802
7803                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
7804
```

```
7805  051210  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
7806  051216  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
7807  051222  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7808  051230  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7809  051236  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
7810  051244  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7811  051250  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7812  051256  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7813
7814                                          ;SEIZE THE DRIVE THROUGH PORT B
7815
7816  051264  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7817  051272  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
7818  051300  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
7819  051304  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
7820  051312  012760  177777  000014          MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT B
7821  051320  012760  177777  000040          MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT B
7822  051326  012760  177777  000042          MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT B
7823  051334  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7824  051342  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7825  051350  004737  052206                  JSR     PC,TST42B       ;CHECK THE REGISTERS THROUGH PORT A
7826  051354  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7827  051362  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7828  051370  005060  000042                  CLR     RPER3(R0)       ;CLEAR RPER3 ON PORT B
7829  051374  005060  000040                  CLR     RPER2(R0)       ;CLEAR RPER2 ON PORT B
7830  051400  005060  000014                  CLR     RPER1(R0)       ;CLEAR RPER1 ON PORT B
7831  051404  013760  001232  000016          MOV     ASR1,RPAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT B
7832  051412  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7833  051420  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7834  051426  012760  177777  000014          MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT A
7835  051434  012760  177777  000040          MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT A
7836  051442  012760  177777  000042          MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT A
7837  051450  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7838  051456  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7839  051464  004737  052206                  JSR     PC,TST42B       ;CHECK THE REGISTERS THROUGH PORT B
7840
7841                                          ;RELEASE THE DRIVE FROM PORT B
7842
7843  051470  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7844  051476  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7845  051504  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
7846
7847                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
7848
7849  051512  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
7850  051516  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7851  051524  012737  000012  001122          MOV     #RPDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
7852  051532  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
7853  051536  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7854  051544  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7855  051552  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
7856  051560  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7857  051566  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7858  051574  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
7859  051602  001404                          BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
7860  051604  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
```

```
7861  051610  001401                        BEQ     66$             ;BR IF ZERO
7862  051612  104031                        ERROR   31              ;REPORT DRIVE IN NEUTRAL
7863  051614  013737  001164  001126  66$:  MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
7864  051622  013737  001224  001234        MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
7865  051630  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
7866  051636  001401                        BEQ     67$             ;BR IF OK
7867  051640  104027                        ERROR   27              ;REPORT REGISTER ERROR
7868  051642  000240                  67$:  NOP
7869
7870                                        ;RELEASE THE DRIVE FROM PORT A
7871
7872  051644  113760  001224  000010        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7873  051652  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7874  051660  012760  000013  000000        MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7875
7876                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7877
7878  051666  005037  001250                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7879  051672  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7880  051700  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7881  051704  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7882  051712  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7883  051720  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7884  051726  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7885  051734  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7886  051742  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7887  051750  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7888  051756  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7889  051764  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7890  051772  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7891  052000  001006                        BNE     68$             ;BR IF NOT
7892  052002  005737  001164                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7893  052006  001045                        BNE     70$             ;BR IF NOT
7894  052010  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7895  052012  000137  052176                JMP     72$             ;BYPASS THE REST OF THE CHECKS
7896  052016  013737  001170  001126  68$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7897  052024  013737  001226  001234        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7898  052032  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7899  052040  005737  001164                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7900  052044  001414                        BEQ     69$             ;BR IF ZERO
7901  052046  013737  001224  001234        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7902  052054  013737  001172  001126        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7903  052062  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7904  052070  005737  001166                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7905  052074  001012                        BNE     70$             ;BR IF NOT
7906  052076  012737  177777  001250  69$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7907  052104  012760  000011  000000        MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7908  052112  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7909  052120  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
7910  052122  013737  001170  001126  70$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7911  052130  013737  001224  001234        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7912  052136  023737  001124  001170        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7913  052144  001401                        BEQ     71$             ;BR IF OK FROM PORT A.
7914  052146  104007                        ERROR   7               ;REPORT ERROR
7915  052150  013737  001172  001126  71$:  MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7916  052156  013737  001226  001234        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
```

```
7917  052164  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7918  052172  001401                          BEQ     72$             ;BR IF OK
7919  052174  104007                          ERROR   7               ;REPORT ERROR
7920  052176  000240                  72$:    NOP
7921  052200  000004                          SCOPE                   ;LOOP ?
7922  052202  000137  052540                  JMP     TST43           ;GO TO THE NEXT TEST
7923
7924                                          ;CHECK THE REGISTERS ON THE SELECTED PORT
7925
7926  052206                          TST42B:
7927  052206  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7928  052212  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
7929  052220  012737  000014  001122          MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7930  052226  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7931  052232  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7932  052236  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7933  052244  001403                          BEQ     64$             ;BR IF OK
7934  052246  104006                          ERROR   6               ;TYPE MESSAGE 6
7935  052250  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7936  052254  016037  000000  001126  64$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7937  052262  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
7938  052270  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
7939  052274  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
7940  052302  001404                          BEQ     65$             ;BR IF NOT
7941  052304  104011                          ERROR   11              ;REPORT THE ERROR
7942  052306  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
7943  052314  000240                  65$:    NOP
7944  052316  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7945  052322  016037  000040  001126          MOV     RPER2(R0),$BDDAT  ;GET CONTENTS OF RPER2
7946  052330  012737  000040  001122          MOV     #RPER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7947  052336  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7948  052342  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7949  052346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7950  052354  001403                          BEQ     66$             ;BR IF OK
7951  052356  104006                          ERROR   6               ;TYPE MESSAGE 6
7952  052360  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7953  052364  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7954  052372  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
7955  052400  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
7956  052404  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
7957  052412  001404                          BEQ     67$             ;BR IF NOT
7958  052414  104011                          ERROR   11              ;REPORT THE ERROR
7959  052416  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
7960  052424  000240                  67$:    NOP
7961  052426  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
7962  052432  016037  000042  001126          MOV     RPER3(R0),$BDDAT  ;GET CONTENTS OF RPER3
7963  052440  012737  000042  001122          MOV     #RPER3,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
7964  052446  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
7965  052452  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
7966  052456  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
7967  052464  001403                          BEQ     68$             ;BR IF OK
7968  052466  104006                          ERROR   6               ;TYPE MESSAGE 6
7969  052470  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
7970  052474  016037  000000  001126  68$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
7971  052502  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
7972  052510  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
```

```
7973  052514  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
7974  052522  001404                          BEQ     69$             ;BR IF NOT
7975  052524  104011                          ERROR   11              ;REPORT THE ERROR
7976  052526  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
7977  052534  000240                  69$:    NOP
7978  052536  000207                          RTS     PC              ;RETURN
7979
7980
7981                                    ;;*****************************************************************
7982                                    ;*TEST 43       TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
7983                                    ;*
7984                                    ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
7985                                    ;*
7986                                    ;*  A.  SET THE ATTENTION BIT FOR PORT 'A'.
7987                                    ;*
7988                                    ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
7989                                    ;*
7990                                    ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
7991                                    ;*      FOR THE DRIVE IS SET.
7992                                    ;*
7993                                    ;;*****************************************************************
7994  052540                            TST43:
7995  052540  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
7996  052544  001406                          BEQ     2$              ;BR IF NOT
7997  052546  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
7998  052550  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
7999  052554  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
8000  052562  112737  000043  001102  2$:     MOVB    #43,$TSTNM      ;TEST NUMBER
8001  052570  012737  052612  001106          MOV     #TEST43,$LPADR  ;LOAD LOOP ON TEST ADDRESS
8002  052576  012737  052612  001110          MOV     #TEST43,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
8003  052604  012737  000031  001176          MOV     #25.,$TIMES     ;;DO 25. ITERATIONS
8004  052612  012706  001100          TEST43: MOV     #STACK,SP       ;LOAD THE STACK POINTER
8005
8006                                    ;CLEAR ATTENTION BITS FOR BOTH PORTS
8007
8008  052616  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
8009  052624  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
8010  052630  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
8011  052636  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8012  052644  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
8013  052652  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
8014  052656  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
8015  052664  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8016  052672  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
8017  052700  012760  177777  000014          MOV     #-1,RPER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT A
8018  052706  005060  000014                  CLR     RPER1(R0)       ;CLEAR THE ERRORS
8019  052712  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
8020  052720  005760  000012          1$:     TST     RPDS1(R0)       ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
8021  052724  001775                          BEQ     1$              ;BR IF STILL SEIZED BY PORT A
8022  052726  012737  000016  001122          MOV     #RPAS,$BDADR    ;FORM ADDRESS OF ATTN REG IF ERROR
8023  052734  060037  001122                  ADD     R0,$BDADR       ;ADD THE ADDRESS BASE
8024  052740  013737  001232  001124          MOV     ASR1,$GDDAT     ;GOOD DATA FOR ERROR MESSAGE
8025  052746  013737  001232  001166          MOV     ASR1,$TMP1      ;MAKE DATA COMPARE MASK
8026  052754  005137  001166                  COM     $TMP1           ;COMPLEMENT IT
8027  052760  012737  053014  001110          MOV     #2$,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
8028  052766  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
```

```
8029  052774  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8030  053002  013737  001226  001236          MOV     PORTB,SEIZPT     ;'SEIZED' PORT ADDRESS
8031  053010  005060  000012                  CLR     RPDS1(R0)        ;SEIZE THE DRIVE THROUGH PORT B
8032  053014  016037  000016  001126  2$:     MOV     RPAS(R0),$BDDAT  ;GET THE CONTENTS OF THE ATTENTION REG
8033  053022  013737  001126  001164          MOV     $BDDAT,$TMP0     ;PUT CONTENTS INTO WORKING LOCATION
8034  053030  043737  001166  001164          BIC     $TMP1,$TMP0      ;CLEAR OTHER BITS
8035  053036  023737  001124  001164          CMP     $GDDAT,$TMP0     ;SEE IF ATTN BIT FOR DRIVE SET
8036  053044  001401                          BEQ     3$               ;BR IF SET
8037  053046  104053                          ERROR   53               ;REPORT THE ERROR
8038  053050                          3$:
8039
8040                                          ;RELEASE THE DRIVE FROM PORT B
8041
8042  053050  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
8043  053056  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8044  053064  012760  000013  000000          MOV     #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B
8045
8046                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8047
8048  053072  005037  001250                  CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8049  053076  012737  000012  001122          MOV     #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8050  053104  060037  001122                  ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
8051  053110  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8052  053116  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
8053  053124  016037  000012  001170          MOV     RPDS1(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8054  053132  013737  001170  001164          MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
8055  053140  042737  100100  001164          BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8056  053146  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
8057  053154  016037  000012  001172          MOV     RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8058  053162  013737  001172  001166          MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
8059  053170  042737  100100  001166          BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8060  053176  023737  001164  001166          CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8061  053204  001006                          BNE     64$              ;BR IF NOT
8062  053206  005737  001164                  TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8063  053212  001045                          BNE     66$              ;BR IF NOT
8064  053214  104046                          ERROR   46               ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8065  053216  000137  053416                  JMP     68$              ;BYPASS THE REST OF THE CHECKS
8066  053222  013737  001170  001126  64$:     MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8067  053230  013737  001226  001234          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8068  053236  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
8069  053244  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
8070  053250  001414                          BEQ     65$              ;BR IF ZERO
8071  053252  013737  001224  001234          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8072  053260  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
8073  053266  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
8074  053274  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
8075  053300  001012                          BNE     66$              ;BR IF NOT
8076  053302  012737  177777  001250  65$:     MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
8077  053310  012760  000011  000000          MOV     #11,RPCS1(R0)    ;CLEAR THE DRIVE
8078  053316  012760  000013  000000          MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
8079  053324  104026                          ERROR   26               ;TYPE ERROR MESSAGE 26
8080  053326  013737  001170  001126  66$:     MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8081  053334  013737  001224  001234          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
8082  053342  042737  100000  001170          BIC     #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
8083  053350  023737  001124  001170          CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
8084  053356  001401                          BEQ     67$              ;BR IF OK FROM PORT A.
```

```
                                                ERROR   7               ;REPORT ERROR
                                        67$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                                MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
                                                BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
                                                CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
                                                BEQ     68$             ;BR IF OK
                                                ERROR   7               ;REPORT ERROR
                                        68$:    NOP
                                                SCOPE                   ;LOOP ?

                                        ;;*****************************************************************
                                        ;*TEST 44       TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
                                        ;*
                                        ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
                                        ;*
                                        ;*  A.  SET THE ATTENTION BIT FOR PORT 'B'.
                                        ;*
                                        ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                                        ;*
                                        ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
                                        ;*      FOR THE DRIVE IS SET.
                                        ;*
                                        ;;*****************************************************************
                                        TST44:
                                                TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
                                                BEQ     2$              ;BR IF NOT
                                                BPL     1$              ;BR IF JUST ENTERED TEST
                                                JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
                                        1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
                                        2$:     MOVB    #44,$TSTNM      ;TEST NUMBER
                                                MOV     #TEST44,$LPADR  ;LOAD LOOP ON TEST ADDRESS
                                                MOV     #TEST44,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
                                                MOV     #25.,$TIMES     ;;DO 25. ITERATIONS
                                        TEST44: MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

                                                MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
                                                CLR     RPDS1(R0)       ;SEIZE THE DRIVE
                                                MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
                                                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                                MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
                                                CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
                                                MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
                                                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                                MOV     #-1,RPER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT B
                                                CLR     RPER1(R0)       ;CLEAR THE ERRORS
                                                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
                                        1$:     TST     RPDS1(R0)       ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
                                                BEQ     1$              ;BR IF STILL SEIZED BY PORT B
                                                MOV     #RPAS,$BDADR    ;FORM ADDRESS OF ATTN REG IF ERROR
                                                ADD     R0,$BDADR       ;ADD THE ADDRESS BASE
                                                MOV     ASR1,$GDDAT     ;GOOD DATA FOR ERROR MESSAGE
                                                MOV     ASR1,$TMP1      ;MAKE DATA COMPARE MASK
                                                COM     $TMP1           ;COMPLEMENT IT
```

```
8141  053642  012737  053676  001110          MOV     #2$,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
8142  053650  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
8143  053656  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8144  053664  013737  001224  001236          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
8145  053672  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT A
8146  053676  016037  000016  001126  2$:     MOV     RPAS(R0),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
8147  053704  013737  001126  001164          MOV     $BDDAT,$TMP0    ;PUT CONTENTS INTO WORKING LOCATION
8148  053712  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR OTHER BITS
8149  053720  023737  001124  001164          CMP     $GDDAT,$TMP0    ;SEE IF ATTN BIT FOR DRIVE SET
8150  053726  001401                          BEQ     3$              ;BR IF SET
8151  053730  104053                          ERROR   53              ;REPORT THE ERROR
8152  053732                          3$:
8153
8154                                          ;RELEASE THE DRIVE FROM PORT A
8155
8156  053732  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
8157  053740  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8158  053746  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8159
8160                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8161
8162  053754  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8163  053760  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8164  053766  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8165  053772  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8166  054000  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8167  054006  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8168  054014  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8169  054022  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8170  054030  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8171  054036  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8172  054044  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8173  054052  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8174  054060  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8175  054066  001006                          BNE     64$             ;BR IF NOT
8176  054070  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8177  054074  001045                          BNE     66$             ;BR IF NOT
8178  054076  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8179  054100  000137  054300                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
8180  054104  013737  001170  001126  64$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8181  054112  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8182  054120  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8183  054126  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8184  054132  001414                          BEQ     65$             ;BR IF ZERO
8185  054134  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8186  054142  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8187  054150  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8188  054156  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8189  054162  001012                          BNE     66$             ;BR IF NOT
8190  054164  012737  177777  001250  65$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8191  054172  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
8192  054200  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8193  054206  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
8194  054210  013737  001170  001126  66$:     MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
8195  054216  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
8196  054224  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
```

```
8197  054232  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
8198  054240  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
8199  054242  104007                          ERROR   7               ;REPORT ERROR
8200  054244  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
8201  054252  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
8202  054260  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
8203  054266  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
8204  054274  001401                          BEQ     68$             ;BR IF OK
8205  054276  104007                          ERROR   7               ;REPORT ERROR
8206  054300  000240                  68$:    NOP
8207  054302  000004                          SCOPE                   ;LOOP ?
8208  054304  000137  056224                  JMP     $EOP            ;GO TO END OF TEST
8209
8210                                  ;;**************************************************************
8211
8212                                  .SBTTL  *** SPECIAL TESTS FOR THE M7775 ('DP') BOARD ***
8213
8214                                  ;;**************************************************************
8215
8216
8217                                  ;;**************************************************************
8218                                  ;*TEST 45       TEST NO TIMEOUT THROUGH PORT 'A'
8219                                  ;*
8220                                  ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
8221                                  ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
8222                                  ;*
8223                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
8224                                  ;*
8225                                  ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
8226                                  ;*
8227                                  ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B'.  VERIFY THAT THE DRIVE
8228                                  ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
8229                                  ;*      SET FOR PORT 'B'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
8230                                  ;*
8231                                  ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
8232                                  ;*      BEEN RELEASED.
8233                                  ;*
8234                                  ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
8235                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
8236                                  ;*
8237                                  ;;**************************************************************
8238  054310
8239  054310  005737  001274          TST45:  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
8240  054314  001406                          BEQ     2$              ;BR IF NOT
8241  054316  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
8242  054320  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
8243  054324  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
8244  054332  112737  000045  001102  2$:     MOVB    #45,$TSTNM      ;TEST NUMBER
8245  054340  012737  054362  001106          MOV     #TEST45,$LPADR  ;LOAD LOOP ON TEST ADDRESS
8246  054346  012737  054362  001110          MOV     #TEST45,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
8247  054354  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
8248  054362  012706  001100          TEST45: MOV     #STACK,SP       ;LOAD THE STACK POINTER
8249
8250                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
8251
8252  054366  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
```

```
8253  054374  005060  000012                  CLR     RPDS1(R0)          ;SEIZE THE DRIVE
8254  054400  012760  000011  000000          MOV     #11,RPCS1(R0)      ;ISSUE DRIVE CLEAR
8255  054406  012760  000013  000000          MOV     #13,RPCS1(R0)      ;RELEASE THE DRIVE
8256  054414  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT #B
8257  054422  005060  000012                  CLR     RPDS1(R0)          ;SEIZE THE DRIVE THROUGH PORT 'B'
8258  054426  012760  000011  000000          MOV     #11,RPCS1(R0)      ;ISSUE DRIVE CLEAR
8259  054434  012760  000013  000000          MOV     #13,RPCS1(R0)      ;RELEASE THE DRIVE
8260
8261                                  ;;**************************************************************
8262
8263                                          ;SEIZE THE DRIVE THROUGH PORT B
8264
8265  054442  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
8266  054450  013737  001226  001236          MOV     PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
8267  054456  005060  000012                  CLR     RPDS1(R0)          ;WRITE RPDS1
8268  054462  013737  001224  001240          MOV     PORTA,OPPRT        ;'OPPOSITE' PORT ADDRESS
8269  054470  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
8270  054476  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8271
8272                                  ;;**************************************************************
8273                                  ;SET REQUEST THROUGH PORT A
8274
8275  054504  005060  000012                  CLR     RPDS1(R0)          ;SET REQUEST FOR PORT A
8276  054510  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
8277  054516  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8278
8279                                  ;;**************************************************************
8280                                  ;RELEASE THE DRIVE THROUGH PORT B
8281
8282  054524  012760  000013  000000          MOV     #13,RPCS1(R0)      ;RELEASE DRIVE THROUGH PORT B
8283
8284                                  ;;**************************************************************
8285                                  ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8286
8287  054532  013737  001260  001254          MOV     TIMEAP,WATCH       ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8288
8289                                  ;;**************************************************************
8290                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
8291
8292  054540  005037  001244                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
8293  054544  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8294  054552  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8295  054560  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
8296  054564  005037  001124                  CLR     $GDDAT             ;WHAT REGISTER SHOULD BE
8297  054570  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
8298  054576  001403                          BEQ     66$                ;BR IF OK
8299  054600  104031                          ERROR   31                 ;TYPE MESSAGE 31
8300  054602  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
8301  054606  000240                  66$:    NOP
8302  054610  005737  001244                  TST     CKERR              ;REGISTER OK ?
8303  054614  001402                          BEQ     .+6                ;BR IF OK
8304  054616  000137  055254                  JMP     1$                 ;BYPASS REST OF TEST IF NOT
8305  054622  005737  001254                  TST     WATCH              ;WATCH EQUAL ZERO ?
8306  054626  001375                          BNE     .-4                ;BR IF NOT
8307
8308                                  ;;**************************************************************
```

```
8309                                    ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
8310
8311  054630  013737  001224  001234            MOV     PORTA,PTNBR     ;PORT NUMBER FOR TYPEOUT
8312  054636  005037  001244                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8313  054642  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8314  054650  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8315  054656  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8316  054662  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8317  054666  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8318  054674  001403                            BEQ     68$             ;BR IF OK
8319  054676  104035                            ERROR   35              ;TYPE MESSAGE 35
8320  054700  005137  001244                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8321  054704  000240                    68$:    NOP
8322  054706  005737  001244                    TST     CKERR           ;REGISTER OK ?
8323  054712  001402                            BEQ     .+6             ;BR IF OK
8324  054714  000137  055254                    JMP     1$              ;BYPASS REST OF TEST IF NOT
8325
8326                                    ;;**************************************************************
8327
8328                                            ;RELEASE THE DRIVE FROM PORT A
8329
8330  054720  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
8331  054726  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8332  054734  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8333
8334                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8335
8336  054742  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8337  054746  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8338  054754  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8339  054760  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8340  054766  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8341  054774  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8342  055002  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8343  055010  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8344  055016  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8345  055024  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8346  055032  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8347  055040  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8348  055046  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8349  055054  001006                            BNE     70$             ;BR IF NOT
8350  055056  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8351  055062  001045                            BNE     72$             ;BR IF NOT
8352  055064  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8353  055066  000137  055252                    JMP     74$             ;BYPASS THE REST OF THE CHECKS
8354  055072  013737  001170  001126    70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8355  055100  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8356  055106  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8357  055114  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8358  055120  001414                            BEQ     71$             ;BR IF ZERO
8359  055122  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8360  055130  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8361  055136  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8362  055144  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8363  055150  001012                            BNE     72$             ;BR IF NOT
8364  055152  012737  177777  001250    71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
```

```
 8365  055160  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 8366  055166  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8367  055174  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 8368  055176  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 8369  055204  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8370  055212  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8371  055220  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
 8372  055222  104007                          ERROR   7               ;REPORT ERROR
 8373  055224  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 8374  055232  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8375  055240  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8376  055246  001401                          BEQ     74$             ;BR IF OK
 8377  055250  104007                          ERROR   7               ;REPORT ERROR
 8378  055252  000240                  74$:    NOP
 8379
 8380  055254  000004                  1$:     SCOPE                   ;LOOP ?
 8381
 8382                                          ;;*****************************************************************
 8383                                          ;*TEST 46       TEST NO TIMEOUT THROUGH PORT 'B'
 8384                                          ;*
 8385                                          ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
 8386                                          ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
 8387                                          ;*
 8388                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 8389                                          ;*
 8390                                          ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
 8391                                          ;*
 8392                                          ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A'.  VERIFY THAT THE DRIVE
 8393                                          ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
 8394                                          ;*      SET FOR PORT 'A'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
 8395                                          ;*
 8396                                          ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
 8397                                          ;*      BEEN RELEASED.
 8398                                          ;*
 8399                                          ;*  E.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
 8400                                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 8401                                          ;*
 8402                                          ;*****************************************************************
 8403  055256                          TST46:
 8404  055256  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 8405  055262  001406                          BEQ     2$              ;BR IF NOT
 8406  055264  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 8407  055266  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8408  055272  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 8409  055300  112737  000046  001102  2$:     MOVB    #46,$TSTNM      ;TEST NUMBER
 8410  055306  012737  055330  001106          MOV     #TEST46,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8411  055314  012737  055330  001110          MOV     #TEST46,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 8412  055322  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 8413  055330  012706  001100          TEST46: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 8414
 8415                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 8416
 8417  055334  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 8418  055342  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 8419  055346  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8420  055354  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
```

```
 8421  055362  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 8422  055370  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 8423  055374  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8424  055402  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8425
 8426                                  ;;*****************************************************************
 8427
 8428                                          ;SEIZE THE DRIVE THROUGH PORT A
 8429
 8430  055410  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 8431  055416  013737  001224  001236          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 8432  055424  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
 8433  055430  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
 8434  055436  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 8435  055444  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8436
 8437                                  ;;*****************************************************************
 8438                                  ;SET REQUEST THROUGH PORT B
 8439
 8440  055452  005060  000012                  CLR     RPDS1(R0)       ;SET REQUEST FOR PORT B
 8441  055456  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 8442  055464  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8443
 8444                                  ;;*****************************************************************
 8445                                  ;RELEASE THE DRIVE THROUGH PORT A
 8446
 8447  055472  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE DRIVE THROUGH PORT A
 8448
 8449                                  ;;*****************************************************************
 8450                                  ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
 8451
 8452  055500  013737  001266  001254          MOV     TIMEBP,WATCH    ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
 8453
 8454                                  ;;*****************************************************************
 8455                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
 8456
 8457  055506  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 8458  055512  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 8459  055520  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 8460  055526  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 8461  055532  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 8462  055536  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 8463  055544  001403                          BEQ     66$             ;BR IF OK
 8464  055546  104031                          ERROR   31              ;TYPE MESSAGE 31
 8465  055550  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 8466  055554  000240                  66$:    NOP
 8467  055556  005737  001244                  TST     CKERR           ;REGISTER OK ?
 8468  055562  001402                          BEQ     .+6             ;BR IF OK
 8469  055564  000137  056222                  JMP     1$              ;BYPASS REST OF TEST IF NOT
 8470  055570  005737  001254                  TST     WATCH           ;WATCH EQUAL ZERO ?
 8471  055574  001375                          BNE     .-4             ;BR IF NOT
 8472
 8473                                  ;;*****************************************************************
 8474                                  ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
 8475
 8476  055576  013737  001226  001234          MOV     PORTB,PTNBR     ;PORT NUMBER FOR TYPEOUT
```

```
8477  055604  005037  001244                    CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8478  055610  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8479  055616  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8480  055624  060037  001122                    ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8481  055630  005037  001124                    CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8482  055634  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8483  055642  001403                            BEQ     68$             ;BR IF OK
8484  055644  104035                            ERROR   35              ;TYPE MESSAGE 35
8485  055646  005137  001244                    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8486  055652  000240                    68$:    NOP
8487  055654  005737  001244                    TST     CKERR           ;REGISTER OK ?
8488  055660  001402                            BEQ     .+6             ;BR IF OK
8489  055662  000137  056222                    JMP     1$              ;BYPASS REST OF TEST IF NOT
8490
8491                                    ;;*****************************************************************
8492
8493                                            ;RELEASE THE DRIVE FROM PORT B
8494
8495  055666  113760  001226  000010            MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
8496  055674  013737  001226  001234            MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8497  055702  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8498
8499                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8500
8501  055710  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8502  055714  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8503  055722  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8504  055726  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8505  055734  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8506  055742  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8507  055750  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8508  055756  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8509  055764  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8510  055772  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8511  056000  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8512  056006  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8513  056014  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8514  056022  001006                            BNE     70$             ;BR IF NOT
8515  056024  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8516  056030  001045                            BNE     72$             ;BR IF NOT
8517  056032  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8518  056034  000137  056220                    JMP     74$             ;BYPASS THE REST OF THE CHECKS
8519  056040  013737  001170  001126    70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8520  056046  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8521  056054  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8522  056062  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8523  056066  001414                            BEQ     71$             ;BR IF ZERO
8524  056070  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8525  056076  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
8526  056104  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8527  056112  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
8528  056116  001012                            BNE     72$             ;BR IF NOT
8529  056120  012737  177777  001250    71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
8530  056126  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
8531  056134  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8532  056142  104026                            ERROR   26              ;TYPE ERROR MESSAGE 26
```

```
 8533  056144  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 8534  056152  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8535  056160  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8536  056166  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
 8537  056170  104007                          ERROR   7               ;REPORT ERROR
 8538  056172  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 8539  056200  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8540  056206  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8541  056214  001401                          BEQ     74$             ;BR IF OK
 8542  056216  104007                          ERROR   7               ;REPORT ERROR
 8543  056220  000240                  74$:    NOP
 8544
 8545  056222  000004                  1$:     SCOPE                   ;LOOP ?
 8546
 8547                                  .SBTTL  END OF PASS ROUTINE
 8548
 8549                                  ;;*************************************************************
 8550                                  ;*INCREMENT THE PASS NUMBER ($PASS)
 8551                                  ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
 8552                                  ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
 8553                                  ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
 8554                                  ;*IF THERES A MONITOR GO TO IT
 8555                                  ;*IF THERE ISN'T JUMP TO TST1AA
 8556
 8557  056224                  $EOP:
 8558  056224  005737  001274                  TST     KYBCTL          ;ENTERED TEST VIA KEYBOARD COMMAND ?
 8559  056230  001402                          BEQ     .+6             ;BR IF NOT
 8560  056232  000137  002602                  JMP     EXEC            ;RETURN TO KEYBOARD CONTROL
 8561  056236  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 8562  056242  005037  001176                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 8563  056246  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 8564  056252  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 8565  056260  005327                          DEC     (PC)+           ;;LOOP?
 8566  056262  000001          $EOPCT: .WORD   1
 8567  056264  003063                          BGT     $DOAGN          ;;YES
 8568  056266  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 8569  056270  000001          $ENDCT: .WORD   1
 8570  056272  056262                          $EOPCT
 8571  056274  104401  056302                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 8572  056300  000407                          BR      64$             ;;GET OVER THE ASCIZ
 8573                                  ;;65$:  .ASCIZ  <12><15>/END PASS #/
 8574  056320                  64$:
 8575  056320  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 8576                                                                  ;;TYPE PASS NUMBER
 8577  056324  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8578  056326  104401  056334                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 8579  056332  000421                          BR      66$             ;;GET OVER THE ASCIZ
 8580                                  ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
 8581  056376                  66$:
 8582  056376  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
 8583                                                                  ;;TOTAL NUMBER OF ERRORS
 8584  056402  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8585  056404  104401  001207                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
 8586  056410  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
 8587  056414  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 8588  056420  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
```

```
 8589  056422  000005                          RESET                   ;;CLEAR THE WORLD
 8590  056424  004710                $ENDAD:   JSR     PC,(R0)         ;;GO TO MONITOR
 8591  056426  000240                          NOP                     ;;SAVE ROOM
 8592  056430  000240                          NOP                     ;;FOR
 8593  056432  000240                          NOP                     ;;ACT11
 8594  056434                        $DOAGN:
 8595  056434  000137                          JMP     @(PC)+          ;;RETURN
 8596  056436  003064                $RTNAD:   .WORD   TST1AA
 8597  056440     377      377  000  $ENULL:   .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 8598          056444                         .EVEN
 8599
 8600                                ;;*****************************************************************
 8601
 8602                                .SBTTL  *** SUBROUTINES ***
 8603
 8604                                ;;*****************************************************************
 8605
 8606
 8607                                ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
 8608                                ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
 8609
 8610  056444  012737  056514  000004  CKCLK:   MOV     #CKCLK1,@#ERRVEC  ;SET UP VECTOR FOR CLOCK CHECK
 8611  056452  005037  000006                  CLR     @#ERRVEC+2        ;NEW PSW
 8612  056456  005777  122530                  TST     @$LKCSR         ;CHECK FOR KW11-P
 8613  056462  013701  001216                  MOV     $LPVEC,R1       ;KW11-P VECTOR ADDRESS
 8614  056466  012721  056576                  MOV     #CLOCK,(R1)+    ;SET UP KW11-P VECTOR
 8615  056472  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
 8616  056476  012777  177777  122510          MOV     #-1,@$LKCSB     ;LOAD COUNTER BUFFER WITH 1'S
 8617  056504  012777  000135  122500          MOV     #135,@$LKCSR    ;SET CLOCK - CNT UP, 16MS, CONT INT
 8618  056512  000425                          BR      CKCLK3
 8619  056514  062706  000004          CKCLK1: ADD     #4,SP           ;RESTORE THE STACK POINTER
 8620  056520  012737  056556  000004          MOV     #CKCLK2,@#ERRVEC  ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
 8621  056526  005777  122466                  TST     @$LKS           ;LOOK FOR KW11-L
 8622  056532  013701  001222                  MOV     $LLVEC,R1       ;KW11-L VECTOR ADDRESS
 8623  056536  012721  056576                  MOV     #CLOCK,(R1)+    ;SET UP KW11-L VECTOR
 8624  056542  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
 8625  056546  012777  000100  122444          MOV     #100,@$LKS      ;SET KW11-L INTERRUPT
 8626  056554  000404                          BR      CKCLK3
 8627  056556  062706  000004          CKCLK2: ADD     #4,SP           ;RESTORE THE STACK POINTER
 8628  056562  062716  000002                  ADD     #2,(SP)         ;INCREMENT RETURN, NO CLOCK
 8629  056566  012737  000006  000004  CKCLK3: MOV     #6,@#ERRVEC     ;RESTORE THE ERROR VECTOR
 8630  056574  000207                          RTS     PC
 8631
 8632                                ;ROUTINE TO COUNT CLOCK TICKS
 8633
 8634  056576  062737  000021  001252  CLOCK:  ADD     #17.,TIME       ;ADD 17 MS TO ELAPSED TIME COUNTER
 8635  056604  005737  001254                  TST     WATCH           ;IS WATCH ALREADY ZERO ?
 8636  056610  001406                          BEQ     1$              ;BR IF IT IS
 8637  056612  162737  000021  001254          SUB     #17.,WATCH      ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
 8638  056620  100002                          BPL     1$              ;BR IF NOT MINUS
 8639  056622  005037  001254                  CLR     WATCH           ;CLEAR WATCH DOG COUNTER
 8640  056626  000002                  1$:     RTI                     ;RETURN
 8641
 8642                                ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
 8643
 8644  056630  162706  000004          TOLER:  SUB     #4,SP   ;SETUP STACK
```

```
8645  056634  016616  000004                  MOV     4(SP),(SP)      ;SAVE STACK
8646  05664C  013546                          MOV     @(R5)+,-(SP)    ;GET TIME VALUE
8647  056652  011666  000004                  MOV     (SP),4(SP)      ;MOVE TIME VALUE
8648  056656  011666  000006                  MOV     (SP),6(SP)      ;MOVE VALUE AGAIN
8649  056662  006216                          ASR     (SP)            ;DIVIDE BY 2
8650  056664  006216                          ASR     (SP)            ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
8651  056666  061666  000004                  ADD     (SP),4(SP)      ;CALCULATE UPPER LIMIT FOR TIMEOUT
8652  056672  162666  000004                  SUB     (SP)+,4(SP)     ;CALCULATE LOWER LIMIT FOR TIMEOUT
8653  056676  000205                          RTS     R5              ;RETURN WITH TOLERANCES ON THE STACK
8654
8655                                  ;;*****************************************************************
8656
8657                                  .SBTTL  'SYSMAC' UTILITY ROUTINES
8658
8659                                  ;;*****************************************************************
8660
8661                                  .SBTTL  SCOPE HANDLER ROUTINE
8662
8663                                  ;;*****************************************************************
8664                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8665                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
8666                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8667                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8668                                  ;*SW14=1         LOOP ON TEST
8669                                  ;*SW11=1         INHIBIT ITERATIONS
8670                                  ;*SW09=1         LOOP ON ERROR
8671                                  ;*CALL
8672                                  ;*      SCOPE           ;;SCOPE=IOT
8673
8674  056700                          $SCOPE:
8675  056700  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
8676  056702  032777  040000  122240  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
8677  056710  001101                          BNE     $OVER           ;;YES IF SW14=1
8678                                  ;#####START OF CODE FOR THE XOR TESTER#####
8679  056712  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
8680                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
8681  056714  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
8682  056720  012737  056730  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
8683  056726  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
8684  056732  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
8685  056736  000453                          BR      $SVLAD          ;;GO TO THE NEXT TEST
8686  056740  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
8687  056742  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
8688  056746  000413                          BR      7$              ;;LOOP ON THE PRESENT TEST
8689  056750                          6$:;#####END OF CODE FOR THE XOR TESTER#####
8690  056750  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
8691  056754  001421                          BEQ     3$              ;;BR IF NO
8692  056756  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
8693  056764  101015                          BHI     3$              ;;BR IF NO
8694  056766  032777  001000  122154          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
8695  056774  001404                          BEQ     4$              ;;BR IF NO
8696  056776  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
8697  057004  000443                          BR      $OVER
8698  057006  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
8699  057012  005037  001176                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
8700  057016  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
```

```
8701  057010  032777  004000  122122  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
8702  057016  001011                          BNE     1$              ;;BR IF YES
8703  057020  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
8704  057024  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
8705  057026  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
8706  057032  023737  001176  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
8707  057040  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
8708  057042  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
8709  057050  013737  057120  001176          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
8710  057056  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
8711  057062  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
8712  057066  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
8713  057072  005037  001200                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
8714  057076  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8715  057104  013777  001102  122030  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
8716  057112  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
8717  057116  000002                          RTI                     ;;FIXES PS
8718  057120  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
8719                           .SBTTL  ERROR HANDLER ROUTINE
8720
8721                           ;;**************************************************************
8722                           ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
8723                           ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8724                           ;*AND GO TO $ERRTYP ON ERROR
8725                           ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8726                           ;*SW15=1         HALT ON ERROR
8727                           ;*SW13=1         INHIBIT ERROR TYPEOUTS
8728                           ;*SW10=1         BELL ON ERROR
8729                           ;*CALL
8730                           ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
8731
8732  057122                  $ERROR:
8733  057122  104407                  CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
8734  057124  113737  001102  001242  MOVB    $TSTNM,TSTNUM
8735  057132  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
8736  057136  001775                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
8737  057140  013777  001102  121774  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
8738  057146  032777  002000  121764  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
8739  057154  001402                  BEQ     1$              ;;NO - SKIP
8740  057156  104401  001202          TYPE    ,$BELL          ;;RING BELL
8741  057162  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
8742  057166  011637  001116          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
8743  057172  162737  000002  001116  SUB     #2,$ERRPC
8744  057200  117737  121712  001114  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
8745  057206  032777  020000  121724  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
8746  057214  001004                  BNE     20$             ;;SKIP TYPEOUTS
8747  057216  004737  057254          JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
8748  057222  104401  001207          TYPE    ,$CRLF
8749  057226                  20$:
8750  057226  005777  121706          2$:     TST     @SWR            ;;HALT ON ERROR
8751  057232  100002                  BPL     3$              ;;SKIP IF CONTINUE
8752  057234  000000                  HALT                    ;;HALT ON ERROR!
8753  057236  104407                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
8754  057240                  3$:
8755  057240  022737  056424  000042  CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
8756  057246  001001                  BNE     6$              ;;BRANCH IF NO
```

```
8757  057250  000000                          HALT                    ;;YES
8758  057252                          6$:
8759  057252  000002                          RTI                     ;;RETURN
8760                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
8761
8762                                  ;;**********************************************************************
8763                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
8764                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
8765                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
8766
8767  057254                          $ERRTYP:
8768  057254  104401  001207                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8769  057260  010046                        MOV     R0,-(SP)        ;;SAVE R0
8770  057262  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
8771  057264  153700  001114              BISB    @#$ITEMB,R0
8772  057270  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
8773                                                                ;;TYPE THE PC OF THE ERROR
8774  057272  013746  001116              MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
8775                                                                ;;ERROR ADDRESS
8776  057276  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8777  057300  000445                        BR      10$             ;;GET OUT
8778  057302  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
8779  057304  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
8780  057306  006300                        ASL     R0
8781  057310  006300                        ASL     R0
8782  057312  062700  001304              ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
8783  057316  012037  057326              MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
8784  057322  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
8785  057324  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
8786  057326  000000                2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
8787  057330  104401  001207              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8788  057334  012037  057344        3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
8789  057340  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
8790  057342  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
8791  057344  000000                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
8792  057346  104401  001207              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8793  057352  010146                5$:     MOV     R1,-(SP)        ;;SAVE R1
8794  057354  012001                        MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
8795  057356  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
8796  057360  012000                        MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
8797  057362  105720                6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
8798  057364  001003                        BNE     7$              ;;BR IF DECIMAL
8799  057366  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
8800  057370  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8801  057372  000402                        BR      8$
8802  057374                          7$:
8803  057374  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
8804  057376  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
8805  057400  005711                8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
8806  057402  001403                        BEQ     9$              ;;BR IF NO
8807  057404  104401  057424              TYPE    ,11$            ;;TYPE TWO(2) SPACES
8808  057410  000764                        BR      6$              ;;LOOP
8809
8810  057412  012601                9$:     MOV     (SP)+,R1        ;;RESTORE R1
8811  057414  012600                10$:    MOV     (SP)+,R0        ;;RESTORE R0
8812  057416  104401  001207              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
 8813  057422  000207                        RTS     PC              ;;RETURN
 8814  057424  020040      000       11$:    .ASCIZ  / /             ;;TWO(2) SPACES
 8815          057430                        .EVEN
 8816                                .SBTTL  TYPE ROUTINE
 8817
 8818                                ;;*****************************************************************
 8819                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 8820                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 8821                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 8822                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 8823                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 8824                                ;*
 8825                                ;*CALL:
 8826                                ;*1) USING A TRAP INSTRUCTION
 8827                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 8828                                ;*OR
 8829                                ;*      TYPE
 8830                                ;*      MESADR
 8831                                ;*
 8832
 8833  057430  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 8834  057434  100002                        BPL     1$              ;;BR IF YES
 8835  057436  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
 8836  057440  000407                        BR      3$              ;;LEAVE
 8837  057442  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
 8838  057444  017600  000002              MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 8839  057450  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 8840  057452  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 8841  057454  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 8842  057456  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
 8843  057460  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 8844  057464  000002                        RTI                     ;;RETURN
 8845  057466  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 8846  057472  001430                        BEQ     8$
 8847  057474  122716  000200              CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 8848  057500  001006                        BNE     5$
 8849  057502  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
 8850  057504  104401                        TYPE                    ;;TYPE A CR AND LF
 8851  057506  001207                        $CRLF
 8852  057510  105037  057644              CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 8853  057514  000755                        BR      2$              ;;GET NEXT CHARACTER
 8854  057516  004737  057600        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 8855  057522  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 8856  057526  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
 8857  057530  013746  001154              MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 8858                                                                ;;AND THE NULL CHAR.
 8859  057534  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 8860  057540  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 8861  057542  004737  057600              JSR     PC,$TYPEC       ;;GO TYPE A NULL
 8862  057546  105337  057644              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 8863  057552  000770                        BR      7$              ;;LOOP
 8864
 8865                                ;HORIZONTAL TAB PROCESSOR
 8866
 8867  057554  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 8868  057560  004737  057600        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
```

```
8869  057564  132737  000007  057644          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
8870  057572  001372                          BNE     9$              ;;TAB STOP
8871  057574  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
8872  057576  000724                          BR      2$              ;;GET NEXT CHARACTER
8873  057600  105777  121344          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
8874  057604  100375                          BPL     $TYPEC
8875  057606  116677  000002  121336          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
8876  057614  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
8877  057622  001003                          BNE     1$              ;;BRANCH IF NO
8878  057624  105037  057644                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
8879  057630  000406                          BR      $TYPEX          ;;EXIT
8880  057632  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
8881  057640  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
8882  057642  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
8883  057644  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
8884  057646  000207          $TYPEX: RTS     PC
8885
8886                          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
8887
8888                          ;;**************************************************************
8889                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
8890                          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
8891                          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
8892                          ;*CALL:
8893                          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
8894                          ;*      TYPOS                   ;;CALL FOR TYPEOUT
8895                          ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
8896                          ;*      .BYTE   M               ;;M=1 OR 0
8897                          ;*                                      ;;1=TYPE LEADING ZEROS
8898                          ;*                                      ;;0=SUPPRESS LEADING ZEROS
8899                          ;*
8900                          ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
8901                          ;*$TYPOS OR $TYPOC
8902                          ;*CALL:
8903                          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
8904                          ;*      TYPON                   ;;CALL FOR TYPEOUT
8905                          ;*
8906                          ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
8907                          ;*CALL:
8908                          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
8909                          ;*      TYPOC                   ;;CALL FOR TYPEOUT
8910
8911  057650  017646  000000  $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
8912  057654  116637  000001  060073  MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
8913  057662  112637  060075          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
8914  057666  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
8915  057672  000406                  BR      $TYPON
8916  057674  112737  000001  060073  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
8917  057702  112737  000006  060075          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
8918  057710  112737  000005  060072  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
8919  057716  010346                          MOV     R3,-(SP)        ;;SAVE R3
8920  057720  010446                          MOV     R4,-(SP)        ;;SAVE R4
8921  057722  010546                          MOV     R5,-(SP)        ;;SAVE R5
8922  057724  113704  060075                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
8923  057730  005404                          NEG     R4
8924  057732  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
```

```
 8925  057736  110437  060074                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 8926  057742  113704  060073                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 8927  057746  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 8928  057752  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
 8929  057754  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 8930  057756  000404                         BR      3$              ;;GO DO MSB
 8931  057760  006105                 2$:     ROL     R5              ;;FORM THIS DIGIT
 8932  057762  006105                         ROL     R5
 8933  057764  006105                         ROL     R5
 8934  057766  010503                         MOV     R5,R3
 8935  057770  006103                 3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 8936  057772  105337  060074                 DECB    $OMODE          ;;TYPE THIS DIGIT?
 8937  057776  100016                         BPL     7$              ;;BR IF NO
 8938  060000  042703  177770                 BIC     #177770,R3      ;;GET RID OF JUNK
 8939  060004  001002                         BNE     4$              ;;TEST FOR 0
 8940  060006  005704                         TST     R4              ;;SUPPRESS THIS 0?
 8941  060010  001403                         BEQ     5$              ;;BR IF YES
 8942  060012  005204                 4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 8943  060014  052703  000060                 BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 8944  060020  052703  000040         5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 8945  060024  110337  060070                 MOVB    R3,8$           ;;SAVE FOR TYPING
 8946  060030  104401  060070                 TYPE    ,8$             ;;GO TYPE THIS DIGIT
 8947  060034  105337  060072         7$:     DECB    $OCNT           ;;COUNT BY 1
 8948  060040  003347                         BGT     2$              ;;BR IF MORE TO DO
 8949  060042  002402                         BLT     6$              ;;BR IF DONE
 8950  060044  005204                         INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 8951  060046  000744                         BR      2$              ;;GO DO THE LAST DIGIT
 8952  060050  012605                 6$:     MOV     (SP)+,R5        ;;RESTORE R5
 8953  060052  012604                         MOV     (SP)+,R4        ;;RESTORE R4
 8954  060054  012603                         MOV     (SP)+,R3        ;;RESTORE R3
 8955  060056  016666  000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 8956  060064  012616                         MOV     (SP)+,(SP)
 8957  060066  000002                         RTI                     ;;RETURN
 8958  060070     000                 8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 8959  060071     000                         .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 8960  060072     000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 8961  060073     000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 8962  060074  000000                 $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 8963                                 .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 8964
 8965                                 ;;**************************************************************
 8966                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 8967                                 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 8968                                 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 8969                                 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 8970                                 ;*REPLACED WITH SPACES.
 8971                                 ;*CALL:
 8972                                 ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 8973                                 ;*      TYPDS                   ;;GO TO THE ROUTINE
 8974
 8975  060076                         $TYPDS:
 8976  060076  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8977  060100  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8978  060102  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8979  060104  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8980  060106  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
 8981  060110  012746  020200                      MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 8982  060114  016605  000020                      MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 8983  060120  100004                              BPL     1$              ;;BR IF INPUT IS POS.
 8984  060122  005405                              NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 8985  060124  112766  000055  000001              MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 8986  060132  005000                      1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 8987  060134  012703  060312                      MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 8988  060140  112723  000040                      MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 8989  060144  005002                      2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 8990  060146  016001  060302                      MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 8991  060152  160105                      3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 8992  060154  002402                              BLT     4$              ;;BR IF DONE
 8993  060156  005202                              INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 8994  060160  000774                              BR      3$
 8995  060162  060105                      4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 8996  060164  005702                              TST     R2              ;;CHECK IF BCD DIGIT=0
 8997  060166  001002                              BNE     5$              ;;FALL THROUGH IF 0
 8998  060170  105716                              TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 8999  060172  100407                              BMI     7$              ;;BR IF YES
 9000  060174  106316                      5$:     ASLB    (SP)            ;;MSD?
 9001  060176  103003                              BCC     6$              ;;BR IF NO
 9002  060200  116663  000001  177777              MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 9003  060206  052702  000060                6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 9004  060212  052702  000040                7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 9005  060216  110223                              MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 9006  060220  005720                              TST     (R0)+           ;;JUST INCREMENTING
 9007  060222  020027  000010                      CMP     R0,#10          ;;CHECK THE TABLE INDEX
 9008  060226  002746                              BLT     2$              ;;GO DO THE NEXT DIGIT
 9009  060230  003002                              BGT     8$              ;;GO TO EXIT
 9010  060232  010502                              MOV     R5,R2           ;;GET THE LSD
 9011  060234  000764                              BR      6$              ;;GO CHANGE TO ASCII
 9012  060236  105726                      8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 9013  060240  100003                              BPL     9$              ;;BR IF NO
 9014  060242  116663  177777  177776              MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 9015  060250  105013                      9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 9016  060252  012605                              MOV     (SP)+,R5        ;;POP STACK INTO R5
 9017  060254  012603                              MOV     (SP)+,R3        ;;POP STACK INTO R3
 9018  060256  012602                              MOV     (SP)+,R2        ;;POP STACK INTO R2
 9019  060260  012601                              MOV     (SP)+,R1        ;;POP STACK INTO R1
 9020  060262  012600                              MOV     (SP)+,R0        ;;POP STACK INTO R0
 9021  060264  104401  060312                      TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 9022  060270  016666  000002  000004              MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 9023  060276  012616                              MOV     (SP)+,(SP)
 9024  060300  000002                              RTI                     ;;RETURN TO USER
 9025  060302  023420                      $DTBL:  10000.
 9026  060304  001750                              1000.
 9027  060306  000144                              100.
 9028  060310  000012                              10.
 9029  060312  000004                      $DBLK:  .BLKW   4
 9030                                      .SBTTL  TTY INPUT ROUTINE
 9031
 9032                                      ;;**********************************************************************
 9033                                      .ENABL  LSB
 9034  060322  000000                      $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
 9035  060324  000000                      $TKQIN: .WORD   0               ;;INPUT POINTER
 9036  060326  000000                      $TKQOUT: .WORD  0               ;;OUTPUT POINTER
```

```
                                TTY INPUT ROUTINE

 9037  060330  000001           $TKQSRT: .BLKB  1                ;;TTY KEYBOARD QUEUE
 9038          060331           $TKQEND=.
 9039          060332           .EVEN
 9040
 9041                           ;*TK INITIALIZE ROUTINE
 9042                           ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
 9043                           ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
 9044                           ;
 9045                           ;*CALL:
 9046                           ;*      JSR     PC,$TKINT
 9047                           ;*      RETURN
 9048                           ;
 9049  060332  005037  060322   $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
 9050  060336  012737  060330  060324         MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
 9051  060344  013737  060324  060326         MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
 9052  060352  012737  060402  000060         MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
 9053  060360  012737  000200  000062         MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
 9054  060366  005777  120554                 TST     @$TKB           ;;CLEAR DONE FLAG
 9055  060372  012777  000100  120544         MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
 9056  060400  000207                         RTS     PC              ;;RETURN TO CALLER
 9057
 9058                           ;*TK SERVICE ROUTINE
 9059                           ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
 9060                           ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
 9061                           ;*IT IN THE QUEUE.
 9062                           ;
 9063  060402  117746  120540   $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
 9064  060406  042716  177600           BIC     #^C177,(SP)     ;;STRIP THE JUNK
 9065  060412  021627  000007   1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
 9066  060416  001004                   BNE     2$              ;;BRANCH IF NO
 9067  060420  022737  000176  001140         CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
 9068  060426  001500                   BEQ     6$              ;;GO TO SWR CHANGE
 9069
 9070  060430           2$:
 9071  060430  022737  000001  060322         CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
 9072  060436  001004                   BNE     3$              ;;BRANCH IF NO
 9073  060440  104401  001202                 TYPE    ,$BELL          ;;RING THE TTY BELL
 9074  060444  005726                   TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
 9075  060446  000451                   BR      5$              ;;EXIT
 9076  060450  021627  000023   3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
 9077  060454  001021                   BNE     32$             ;;BRANCH IF NO
 9078  060456  005077  120462           CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9079  060462  005726                   TST     (SP)+           ;;CLEAN CHAR OFF STACK
 9080  060464  105777  120454   31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
 9081  060470  100375                   BPL     31$             ;;LOOP UNTIL ITS THERE
 9082  060472  117746  120450           MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
 9083  060476  042716  177600           BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9084  060502  022627  000021           CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 9085  060506  001366                   BNE     31$             ;;BRANCH IF NO
 9086  060510  012777  000100  120426         MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 9087  060516  000002                   RTI                     ;;RETURN
 9088  060520  005237  060322   32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 9089  060524  021627  000140           CMP     (SP),#140       ;;IS IT UPPER CASE?
 9090  060530  002405                   BLT     4$              ;;BRANCH IF YES
 9091  060532  021627  000175           CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 9092  060536  003002                   BGT     4$              ;;BRANCH IF YES
```

```
                                TTY INPUT ROUTINE

 9093  060540  042716  000040          BIC     #40,(SP)          ;;MAKE IT UPPER CASE
 9094  060544  112677  177554   4$:    MOVB    (SP)+,@$TKQIN     ;;AND PUT IT IN QUEUE
 9095  060550  005237  060324          INC     $TKQIN            ;;UPDATE THE POINTER
 9096  060554  023727  060324  060331  CMP     $TKQIN,#$TKQEND   ;;GO OFF THE END?
 9097  060562  001003                  BNE     5$                ;;BRANCH IF NO
 9098  060564  012737  060330  060324  MOV     #$TKQSRT,$TKQIN   ;;RESET THE POINTER
 9099  060572  000002          5$:     RTI                       ;;RETURN
 9100
 9101                                  ;;*****************************************************************
 9102                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 9103                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
 9104                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
 9105                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 9106  060574  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR        ;;IS THE SOFT-SWR SELECTED
 9107  060602  001104                  BNE     15$               ;;EXIT IF NOT
 9108  060604  105777  120334          TSTB    @$TKS             ;;IS A CHAR WAITING?
 9109  060610  100101                  BPL     15$               ;;IF NOT, EXIT
 9110  060612  117746  120330          MOVB    @$TKB,-(SP)       ;;YES
 9111  060616  042716  177600          BIC     #^C177,(SP)       ;;MAKE IT  7-BIT ASCII
 9112  060622  021627  000007          CMP     (SP),#7           ;;IS IT A CONTROL-G?
 9113  060626  001300                  BNE     2$                ;;IF NOT, PUT IT IN THE TTY QUEUE
 9114                                                            ;;AND EXIT
 9115
 9116                                  ;;*****************************************************************
 9117                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 9118                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 9119                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 9120  060630  123727  001134  000001  6$:     CMPB    $AUTOB,#1         ;;ARE WE RUNNING IN AUTO-MODE?
 9121  060636  001674                  BEQ     2$                ;;BRANCH IF YES
 9122  060640  005726                  TST     (SP)+             ;;CLEAR CONTROL-G OFF STACK
 9123  060642  004737  060332          JSR     PC,$TKINT         ;;FLUSH THE TTY INPUT QUEUE
 9124  060646  005077  120272          CLR     @$TKS             ;;DISABLE TTY KEYBOARD INTERRUPTS
 9125  060652  112737  000001  001135  MOVB    #1,$INTAG         ;;SET INTERRUPT MODE INDICATOR
 9126
 9127  060660  104401  061436          TYPE    ,$CNTLG           ;;ECHO THE CONTROL-G (^G)
 9128  060664  104401  061443  $GTSWR: TYPE    ,$MSWR            ;;TYPE CURRENT CONTENTS
 9129  060670  013746  000176          MOV     $WREG,-(SP)       ;;SAVE SWREG FOR TYPEOUT
 9130  060674  104402                  TYPOC                     ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9131  060676  104401  061454          TYPE    ,$MNEW            ;;PROMPT FOR NEW SWR
 9132  060702  005046          19$:    CLR     -(SP)             ;;CLEAR COUNTER
 9133  060704  005046                  CLR     -(SP)             ;;THE NEW SWR
 9134  060706  105777  120232  7$:     TSTB    @$TKS             ;;CHAR THERE?
 9135  060712  100375                  BPL     7$                ;;IF NOT TRY AGAIN
 9136
 9137  060714  117746  120226          MOVB    @$TKB,-(SP)       ;;PICK UP CHAR
 9138  060720  042716  177600          BIC     #^C177,(SP)       ;;MAKE IT 7-BIT ASCII
 9139
 9140
 9141
 9142  060724  021627  000025  9$:     CMP     (SP),#25          ;;IS IT A CONTROL-U?
 9143  060730  001005                  BNE     10$               ;;BRANCH IF NOT
 9144  060732  104401  061431          TYPE    ,$CNTLU           ;;YES, ECHO CONTROL-U (^U)
 9145  060736  062706  000006  20$:    ADD     #6,SP             ;;IGNORE PREVIOUS INPUT
 9146  060742  000757                  BR      19$               ;;LET'S TRY IT AGAIN
 9147
 9148
```

```
9149  060744  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
9150  060750  001022                          BNE     16$             ;;BRANCH IF NO
9151  060752  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
9152  060756  001403                          BEQ     11$             ;;BRANCH IF YES
9153  060760  016677  000002  120152          MOV     2(SP),@SWR      ;;SAVE NEW SWR
9154  060766  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
9155  060772  104401  001207          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
9156  060776  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
9157  061004  001003                          BNE     15$             ;;BRANCH IF NOT
9158  061006  012777  000100  120130          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
9159  061014  000002          15$:    RTI                             ;;RETURN
9160  061016  004737  057600          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
9161  061022  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
9162  061026  002420                          BLT     18$             ;;BRANCH IF YES
9163  061030  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
9164  061034  003015                          BGT     18$             ;;BRANCH IF YES
9165  061036  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
9166  061042  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
9167  061046  001403                          BEQ     17$             ;;BRANCH IF YES
9168  061050  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
9169  061052  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
9170  061054  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
9171  061056  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
9172  061062  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
9173  061066  000707                          BR      7$              ;;GET THE NEXT ONE
9174  061070  104401  001206          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
9175  061074  000720                          BR      20$             ;;SIMULATE CONTROL-U
9176                                  .DSABL  LSB
9177
9178
9179                                  ;;*****************************************************************
9180                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9181                                  ;*CALL:
9182                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
9183                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
9184                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
9185                                  ;
9186
9187  061076  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
9188  061100  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
9189  061106  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
9190  061112  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
9191  061114  012746  061122                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
9192  061120  000002                          RTI                     ;;POP NEW PC AND PS
9193  061122                          64$:
9194  061122  005737  060322          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
9195  061126  001775                          BEQ     1$
9196  061130  005337  060322                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
9197  061134  117766  177166  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
9198  061142  005237  060326                  INC     $TKQOUT         ;;UPDATE THE POINTER
9199  061146  023727  060326  060331          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
9200  061154  001003                          BNE     2$              ;;BRANCH IF NO
9201  061156  012737  060330  060326          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
9202  061164  000002          2$:     RTI                             ;;RETURN
9203                                  ;;*****************************************************************
9204                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
```

```
 9205                                   ;*CALL:
 9206                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 9207                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 9208                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 9209
 9210  061166  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 9211  061170  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
 9212  061172  012703  061422          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 9213  061176  022703  061431          2$:     CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
 9214  061202  101456                          BLOS    4$              ;;BR IF YES
 9215  061204  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 9216  061206  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 9217  061210  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 9218  061214  001022                          BNE     5$              ;;BR IF NO
 9219  061216  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
 9220  061220  001007                          BNE     6$              ;;BR IF NO
 9221  061222  112737  000134  061420        MOVB    #'\,9$          ;;TYPE A BACK SLASH
 9222  061230  104401  061420                TYPE    ,9$
 9223  061234  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
 9224  061240  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
 9225  061242  020327  061422                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
 9226  061246  103434                          BLO     4$              ;;BR IF YES
 9227  061250  111337  061420                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
 9228  061254  104401  061420                TYPE    ,9$             ;;GO TYPE
 9229  061260  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
 9230  061262  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
 9231  061264  001406                          BEQ     7$              ;;BR IF NO
 9232  061266  112737  000134  061420        MOVB    #'\,9$          ;;TYPE A BACK SLASH
 9233  061274  104401  061420                TYPE    ,9$
 9234  061300  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 9235  061302  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 9236  061306  001003                          BNE     8$              ;;BR IF NO
 9237  061310  104401  061431                TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
 9238  061314  000726                          BR      1$              ;;GO START OVER
 9239  061316  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
 9240  061322  001011                          BNE     3$              ;;BRANCH IF NO
 9241  061324  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 9242  061326  104401  001207                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
 9243  061332  104401  061422                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 9244  061336  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 9245  061340  104401  001206          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 9246  061344  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 9247  061346  111337  061420          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 9248  061352  104401  061420                TYPE    ,9$
 9249  061356  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 9250  061362  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 9251  061364  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 9252  061370  104401  001210                TYPE    ,$LF            ;;TYPE A LINE FEED
 9253  061374  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 9254  061376  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 9255  061400  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 9256  061402  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 9257  061410  012766  061422  000004        MOV     #$TTYIN,4(SP)
 9258  061416  000002                          RTI                     ;;RETURN
 9259  061420     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 9260  061421     000                          .BYTE   0               ;;TERMINATOR
```

```
9261  061422  000007                  $TTYIN: .BLKB   7                       ;;RESERVE 7 BYTES FOR TTY INPUT
9262  061431     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>            ;;CONTROL "U"
9263  061436  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>            ;;CONTROL "G"
9264  061443     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
9265  061450  036440  000040
9266  061454  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
9267  061462  020075     000
9268          061466                  .EVEN
9269                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
9270
9271                                  ;;*****************************************************************
9272                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9273                                  ;*CHANGE IT TO BINARY.
9274                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9275                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
9276                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9277                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9278                                  ;*CALL:
9279                                  ;*      RDOCT                           ;;READ AN OCTAL NUMBER
9280                                  ;*      RETURN HERE                     ;;LOW ORDER BITS ARE ON TOP OF THE STACK
9281                                  ;*                                      ;;HIGH ORDER BITS ARE IN $HIOCT
9282
9283  061466  011646                  $RDOCT: MOV     (SP),-(SP)              ;;PROVIDE SPACE FOR THE
9284  061470  016666  000004  000002          MOV     4(SP),2(SP)             ;;INPUT NUMBER
9285  061476  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
9286  061500  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
9287  061502  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
9288  061504  104411                  1$:     RDLIN                           ;;READ AN ASCIZ LINE
9289  061506  012600                          MOV     (SP)+,R0                ;;GET ADDRESS OF 1ST CHARACTER
9290  061510  010037  061614                  MOV     R0,5$                   ;;AND SAVE IT
9291  061514  005001                          CLR     R1                      ;;CLEAR DATA WORD
9292  061516  005002                          CLR     R2
9293  061520  112046                  2$:     MOVB    (R0)+,-(SP)             ;;PICKUP THIS CHARACTER
9294  061522  001420                          BEQ     3$                      ;;IF ZERO GET OUT
9295  061524  122716  000060                  CMPB    #'0,(SP)                ;;MAKE SURE THIS CHARACTER
9296  061530  003026                          BGT     4$                      ;;IS AN OCTAL DIGIT
9297  061532  122716  000067                  CMPB    #'7,(SP)
9298  061536  002423                          BLT     4$
9299  061540  006301                          ASL     R1                      ;;*2
9300  061542  006102                          ROL     R2
9301  061544  006301                          ASL     R1                      ;;*4
9302  061546  006102                          ROL     R2
9303  061550  006301                          ASL     R1                      ;;*8
9304  061552  006102                          ROL     R2
9305  061554  042716  177770                  BIC     #↑C7,(SP)               ;;STRIP THE ASCII JUNK
9306  061560  062601                          ADD     (SP)+,R1                ;;ADD IN THIS DIGIT
9307  061562  000756                          BR      2$                      ;;LOOP
9308  061564  005726                  3$:     TST     (SP)+                   ;;CLEAN TERMINATOR FROM STACK
9309  061566  010166  000012                  MOV     R1,12(SP)               ;;SAVE THE RESULT
9310  061572  010237  061624                  MOV     R2,$HIOCT
9311  061576  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
9312  061600  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
9313  061602  012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
9314  061604  000002                          RTI                             ;;RETURN
9315  061606  005726                  4$:     TST     (SP)+                   ;;CLEAN PARTIAL FROM STACK
9316  061610  105010                          CLRB    (R0)                    ;;SET A TERMINATOR
```

```
                                READ AN OCTAL NUMBER FROM THE TTY

 9317  061612  104401                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
 9318  061614  000000              5$:       .WORD   0
 9319  061616  104401  001206                TYPE    ,$QUES          ;;"?" "CR" & "LF"
 9320  061622  000730                        BR      1$              ;;TRY AGAIN
 9321  061624  000000              $HIOCT: .WORD   0                 ;;HIGH ORDER BITS GO HERE
 9322                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
 9323
 9324                                ;;******************************************************************
 9325                                ;*SAVE R0-R5
 9326                                ;*CALL:
 9327                                ;*      SAVREG
 9328                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
 9329                                ;*
 9330                                ;*TOP---(+16)
 9331                                ;* +2---(+18)
 9332                                ;* +4---R5
 9333                                ;* +6---R4
 9334                                ;* +8---R3
 9335                                ;*+10---R2
 9336                                ;*+12---R1
 9337                                ;*+14---R0
 9338
 9339  061626                        $SAVREG:
 9340  061626  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9341  061630  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9342  061632  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9343  061634  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9344  061636  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9345  061640  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9346  061642  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 9347  061646  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 9348  061652  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 9349  061656  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 9350  061662  000002                        RTI
 9351
 9352                                ;*RESTORE R0-R5
 9353                                ;*CALL:
 9354                                ;*      RESREG
 9355  061664                        $RESREG:
 9356  061664  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 9357  061670  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 9358  061674  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 9359  061700  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 9360  061704  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9361  061706  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 9362  061710  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9363  061712  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9364  061714  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9365  061716  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9366  061720  000002                        RTI
 9367                                .SBTTL  TRAP DECODER
 9368
 9369                                ;;******************************************************************
 9370                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 9371                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 9372                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
```

```
                                    :*GO TO THAT ROUTINE.
                                    
061722  010046                      $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
061724  016600  000002                      MOV     2(SP),R0        ;;GET TRAP ADDRESS
061730  005740                              TST     -(R0)           ;;BACKUP BY 2
061732  111000                              MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
061734  006300                              ASL     R0              ;;POSITION FOR INDEXING
061736  016000  061756                      MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
061742  000200                              RTS     R0              ;;GO TO ROUTINE
                                    
                                    
                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
                                    
061744  011646                      $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
061746  016666  000004  000002              MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
061754  000002                              RTI                     ;;RESTORE THE PSW
                                    
                                    .SBTTL  TRAP TABLE
                                    
                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                    ;*BY THE "TRAP" INSTRUCTION.
                                    
                                    ;       ROUTINE
                                    ;       -------
061756  061744                      $TRPAD: .WORD   $TRAP2
061760  057430                              $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
061762  057674                              $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
061764  057650                              $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
061766  057710                              $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
061770  060076                              $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
                                    
061772  060664                              $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
                                    
061774  060574                              $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
061776  061076                              $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
062000  061166                              $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
062002  061466                              $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
062004  061626                              $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
062006  061664                              $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
                                    
                                    ;;**************************************************************
                                    
                                    .SBTTL  TELETYPE MESSAGES
                                    
                                    ;;**************************************************************
                                    
062010  005015  046412  044501      TITLE:  .ASCII  <CR><LF><LF>/MAINDEC-11-DZRJE-A/<CR><LF>
062016  042116  041505  030455
062024  026461  055104  045122
062032  026505  006501     012
062037     122  030120  027464              .ASCIZ  @RP04/5/6 DUAL CONTROLLER LOGIC TEST - PART 1@<CR><LF><LF>
062044  027465  020066  052504
062052  046101  041440  047117
062060  051124  046117  042514
062066  020122  047514  044507
062074  020103  042524  052123
```

```
 9429  062102  026440  050040  051101
 9430  062110  020124  006461  005012
 9431  062116     000
 9432  062117     015  042412  052116  ENTERA: .ASCIZ  <CR><LF>/ENTER DRIVE ADDRESS: /
 9433  062124  051105  042040  044522
 9434  062132  042526  040440  042104
 9435  062140  042522  051523  020072
 9436  062146     000
 9437  062147     111  053116  046101  ADRERR: .ASCIZ  /INVALID ADDRESS/<CR><LF>
 9438  062154  042111  040440  042104
 9439  062162  042522  051523  005015
 9440  062170     000
 9441  062171     015  050012  051117  PORTAIS: .ASCIZ <CR><LF>/PORT A ADDRESS IS: /
 9442  062176  020124  020101  042101
 9443  062204  051104  051505  020123
 9444  062212  051511  020072     000
 9445  062217     015  050012  051117  PORTBIS: .ASCIZ <CR><LF>/PORT B ADDRESS IS: /
 9446  062224  020124  020102  042101
 9447  062232  051104  051505  020123
 9448  062240  051511  020072     000
 9449  062245     015  051412  051531  NOCLOCK: .ASCIZ <CR><LF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CR><LF><LF>
 9450  062252  042524  020115  052515
 9451  062260  052123  044040  053101
 9452  062266  020105  046047  020047
 9453  062274  051117  023440  023520
 9454  062302  041440  047514  045503
 9455  062310  005015  000012
 9456  062314  042412  052116  051105  TESTNO: .ASCIZ  <LF>/ENTER TEST #: /
 9457  062322  052040  051505  020124
 9458  062330  035043  000040
 9459  062334  047111  040526  044514  BADNO:  .ASCIZ  /INVALID TEST NUMBER/<CR><LF>
 9460  062342  020104  042524  052123
 9461  062350  047040  046525  042502
 9462  062356  006522  000012
 9463  062362  042440  051122  051117  TSTERR: .ASCIZ  / ERRORS/<CR><LF>
 9464  062370  006523  000012
 9465  062374  005015  052012  042510  ADDRIS: .ASCIZ  <CR><LF><LF>/THE PRESENT ADDRESS OF THE RH11 (RPCS1) IS: /
 9466  062402  050040  042522  042523
 9467  062410  052116  040440  042104
 9468  062416  042522  051523  047440
 9469  062424  020106  044124  020105
 9470  062432  044122  030461  024040
 9471  062440  050122  051503  024461
 9472  062446  044440  035123  000040
 9473  062454  042412  052116  051105  NTRH11: .ASCIZ  <LF>/ENTER NEW RH11 ADDRESS: /
 9474  062462  047040  053505  051040
 9475  062470  030510  020061  042101
 9476  062476  051104  051505  035123
 9477  062504  000040
 9478
 9479                          ;;*****************************************************************
 9480
 9481                          .SBTTL  TEST ERROR MESSAGES
 9482
 9483                          ;;*****************************************************************
 9484
```

```
 9485  062506  051127  047117  020107  EM1:    .ASCIZ  /WRONG DRIVE TYPE/
 9486  062514  051104  053111  020105
 9487  062522  054524  042520     000
 9488
 9489  062527     104  044522  042526  EM2:    .ASCIZ  /DRIVE NOT ON LINE/
 9490  062534  047040  052117  047440
 9491  062542  020116  044514  042516
 9492  062550     000
 9493
 9494  062551     123  051105  040511  EM3:    .ASCIZ  /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
 9495  062556  020114  052516  041115
 9496  062564  051105  051040  040505
 9497  062572  020104  044124  047522
 9498  062600  043525  020110  040505
 9499  062606  044103  050040  051117
 9500  062614  020124  047516  020124
 9501  062622  044124  020105  040523
 9502  062630  042515     000
 9503
 9504  062633     104  044522  042526  EM4:    .ASCIZ  /DRIVE NOT SEIZED BY PORT/
 9505  062640  047040  052117  051440
 9506  062646  044505  042532  020104
 9507  062654  054502  050040  051117
 9508  062662  000124
 9509
 9510  062664  051127  047117  020107  EM5:    .ASCIZ  /WRONG STATUS SEEN BY THE SEIZING PORT/
 9511  062672  052123  052101  051525
 9512  062700  051440  042505  020116
 9513  062706  054502  052040  042510
 9514  062714  051440  044505  044532
 9515  062722  043516  050040  051117
 9516  062730  000124
 9517
 9518  062732  042522  044507  052123  EM6:    .ASCIZ  /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
 9519  062740  051105  041440  047117
 9520  062746  042524  052116  020123
 9521  062754  042527  042522  051440
 9522  062762  042505  020116  054502
 9523  062770  047440  050120  051517
 9524  062776  052111  020105  047520
 9525  063004  052122  026440  042040
 9526  063012  044522  042526  053440
 9527  063020  051501  051440  044505
 9528  063026  042532  000104
 9529
 9530  063032  042522  044507  052123  EM7:    .ASCIZ  /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
 9531  063040  051105  041440  047117
 9532  063046  042524  052116  020123
 9533  063054  051127  047117  020107
 9534  063062  043101  042524  020122
 9535  063070  042522  042514  051501
 9536  063076  020105  051117  052040
 9537  063104  046511  047505  052125
 9538  063112     000
 9539
 9540  063113     122  043505  051511  EM10:   .ASCIZ  /REGISTER CONTENTS WRONG/
```

```
 9541  063120  042524  020122  047503
 9542  063126  052116  047105  051524
 9543  063134  053440  047522  043516
 9544  063142     000
 9545
 9546  063143     103  047117  051124  EM11:   .ASCIZ  /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
 9547  063150  046117  041040  051525
 9548  063156  050040  051101  052111
 9549  063164  020131  051105  047522
 9550  063172  020122  042522  042101
 9551  063200  047111  020107  047111
 9552  063206  044504  040503  042524
 9553  063214  020104  042522  044507
 9554  063222  052123  051105     000
 9555
 9556  063227     104  044522  042526  EM12:   .ASCIZ  /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
 9557  063234  047040  052117  051440
 9558  063242  044505  042532  020104
 9559  063250  054502  042040  044522
 9560  063256  042526  041440  042514
 9561  063264  051101  041440  046517
 9562  063272  040515  042116     000
 9563
 9564  063277     122  040505  044504  EM13:   .ASCIZ  /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
 9565  063304  020116  051120  051505
 9566  063312  052105  042040  042517
 9567  063320  020123  047516  020124
 9568  063326  042523  020124  047526
 9569  063334  052514  042515  053040
 9570  063342  046101  042111  043040
 9571  063350  051117  052040  042510
 9572  063356  050040  051117  000124
 9573
 9574  063364  047526  052514  042515  EM14:   .ASCIZ  /VOLUME VALID SET ON THE WRONG PORT/
 9575  063372  053040  046101  042111
 9576  063400  051440  052105  047440
 9577  063406  020116  044124  020105
 9578  063414  051127  047117  020107
 9579  063422  047520  052122     000
 9580
 9581  063427     101  052124  020116  EM15:   .ASCIZ  /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
 9582  063434  044502  020124  051127
 9583  063442  047117  020107  043101
 9584  063450  042524  020122  044524
 9585  063456  042515  052517  020124
 9586  063464  020055  042522  052521
 9587  063472  051505  020124  047516
 9588  063500  020124  042523  000124
 9589
 9590  063506  052101  047124  041040  EM16:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
 9591  063514  052111  053440  047522
 9592  063522  043516  040440  052106
 9593  063530  051105  051040  046105
 9594  063536  040505  042523  026440
 9595  063544  051040  050505  042525
 9596  063552  052123  051440  052105
```

```
 9597  063560     000
 9598
 9599  063561     101  052124  020116  EM17:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
 9600  063566  044502  020124  051127
 9601  063574  047117  020107  043101
 9602  063602  042524  020122  042522
 9603  063610  042514  051501  020105
 9604  063616  020055  042522  052521
 9605  063624  051505  020124  047516
 9606  063632  020124  042523  000124
 9607
 9608  063640  051104  053111  020105  EM20:   .ASCIZ  /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
 9609  063646  047516  020124  042523
 9610  063654  055111  042105  053440
 9611  063662  042510  020116  052101
 9612  063670  047124  041040  052111
 9613  063676  043040  051117  050040
 9614  063704  051117  020124  046103
 9615  063712  040505  042522  000104
 9616
 9617  063720  051104  053111  020105  EM21:   .ASCIZ  /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
 9618  063726  042523  055111  042105
 9619  063734  053440  042510  020116
 9620  063742  042532  047522  053440
 9621  063750  044522  052124  047105
 9622  063756  044440  020116  052101
 9623  063764  047124  041040  052111
 9624  063772     000
 9625
 9626  063773     104  044522  042526  EM22:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
 9627  064000  047040  052117  044440
 9628  064006  020116  042516  052125
 9629  064014  040522  020114  043101
 9630  064022  042524  020122  044524
 9631  064030  042515  052517  020124
 9632  064036  020055  042522  052521
 9633  064044  051505  020124  047516
 9634  064052  020124  042523  000124
 9635
 9636  064060  044524  042515  052517  EM23:   .ASCIZ  /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
 9637  064066  020124  046103  040505
 9638  064074  042522  020104  044124
 9639  064102  020105  051104  053111
 9640  064110  023505  020123  051105
 9641  064116  047522  020122  044502
 9642  064124  000124
 9643
 9644  064126  042522  042514  051501  EM24:   .ASCIZ  /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
 9645  064134  020105  047503  046515
 9646  064142  047101  020104  042522
 9647  064150  042514  051501  042105
 9648  064156  042040  044522  042526
 9649  064164  053440  052111  020110
 9650  064172  051105  047522  051522
 9651  064200  051440  052105     000
 9652
```

```
 9653  064205     124  046511  047505  EM25:   .ASCIZ  /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
 9654  064212  052125  047440  042516
 9655  064220  051455  047510  020124
 9656  064226  044504  020104  047516
 9657  064234  020124  042522  051124
 9658  064242  043511  042507  000122
 9659
 9660  064250  051104  053111  020105  EM26:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
 9661  064256  047516  020124  047111
 9662  064264  047040  052505  051124
 9663  064272  046101  040440  052106
 9664  064300  051105  051040  046105
 9665  064306  040505  042523  026440
 9666  064314  051040  050505  042525
 9667  064322  052123  047040  052117
 9668  064330  051440  052105     000
 9669
 9670  064335     122  043505  051511  EM27:   .ASCIZ  /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
 9671  064342  042524  020122  051127
 9672  064350  047117  020107  043101
 9673  064356  042524  020122  042522
 9674  064364  042514  051501  020105
 9675  064372  044527  044124  051040
 9676  064400  050505  042525  052123
 9677  064406  051440  052105     000
 9678
 9679  064413     104  044522  042526  EM30:   .ASCIZ  /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
 9680  064420  051440  044505  042532
 9681  064426  020104  054502  051040
 9682  064434  046105  040505  042523
 9683  064442  041440  046517  040515
 9684  064450  042116  044440  051523
 9685  064456  042525  020104  044127
 9686  064464  047105  042040  044522
 9687  064472  042526  044440  020116
 9688  064500  042516  052125  040522
 9689  064506  000114
 9690
 9691  064510  051104  053111  020105  EM31:   .ASCIZ  /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
 9692  064516  047111  047040  052505
 9693  064524  051124  046101  040440
 9694  064532  052106  051105  051040
 9695  064540  046105  040505  042523
 9696  064546  026440  051040  050505
 9697  064554  042525  052123  051440
 9698  064562  052105     000
 9699
 9700  064565     101  052124  020116  EM32:   .ASCIZ  /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
 9701  064572  044502  020124  051127
 9702  064600  047117  020107  043101
 9703  064606  042524  020122  042522
 9704  064614  040503  044514  051102
 9705  064622  052101  020105  047503
 9706  064630  046515  047101  000104
 9707
 9708  064636  051104  053111  020105  EM33:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
```

```
 9709  064644  042522  052524  047122
 9710  064652  042105  052040  020117
 9711  064660  042516  052125  040522
 9712  064666  020114  043111  042040
 9713  064674  044522  042526  041440
 9714  064702  042514  051101  043440
 9715  064710  053111  047105  053440
 9716  064716  044510  042514  042040
 9717  064724  044522  042526  051440
 9718  064732  044505  042532  000104
 9719
 9720  064740  051104  053111  020105  EM34:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
 9721  064746  042522  052524  047122
 9722  064754  042105  052040  020117
 9723  064762  042516  052125  040522
 9724  064770  020114  043111  046440
 9725  064776  051501  041123  051525
 9726  065004  044440  044516  020124
 9727  065012  044507  042526  020116
 9728  065020  044127  046111  020105
 9729  065026  051104  053111  020105
 9730  065034  042523  055111  042105
 9731  065042     000
 9732
 9733  065043     124  046511  047505  EM35:   .ASCIZ  /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
 9734  065050  052125  047440  042516
 9735  065056  051440  047510  020124
 9736  065064  044506  042522  020104
 9737  065072  044527  044124  052517
 9738  065100  020124  042522  044507
 9739  065106  052123  051105  040440
 9740  065114  041503  051505  000123
 9741
 9742  065122  044524  042515  052517  EM36:   .ASCIZ  /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
 9743  065130  020124  040510  020123
 9744  065136  047516  020124  041517
 9745  065144  052503  051122  042105
 9746  065152  053440  052111  044510
 9747  065160  020116  020062  042523
 9748  065166  047503  042116  000123
 9749
 9750  065174  051104  053111  020105  EM37:   .ASCIZ  /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
 9751  065202  051511  047040  047117
 9752  065210  042455  044530  052123
 9753  065216  047105  020124  023450
 9754  065224  042516  023504  041040
 9755  065232  052111  051440  052105
 9756  065240  000051
 9757
 9758  065242  052101  047124  041040  EM40:   .ASCIZ  /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
 9759  065250  052111  043040  051117
 9760  065256  050040  051117  020124
 9761  065264  047516  020124  042522
 9762  065272  042523  020124  054502
 9763  065300  046440  051501  041123
 9764  065306  051525  041440  042514
```

```
 9765  065314  051101     000
 9766
 9767  065317     124  046511  047505  EM41:   .ASCIZ  /TIMEOUT CLEARED THE ATTENTION BIT/
 9768  065324  052125  041440  042514
 9769  065332  051101  042105  052040
 9770  065340  042510  040440  052124
 9771  065346  047105  044524  047117
 9772  065354  041040  052111     000
 9773
 9774  065361     104  044522  042526  EM42:   .ASCIZ  /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
 9775  065366  047040  052117  044440
 9776  065374  020116  042516  052125
 9777  065402  040522  020114  051117
 9778  065410  051440  044505  042532
 9779  065416  020104  043101  042524
 9780  065424  020122  052101  047124
 9781  065432  041040  052111  053440
 9782  065440  044522  052124  047105
 9783  065446     000
 9784
 9785  065447     104  044522  042526  EM43:   .ASCIZ  /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
 9786  065454  044440  020116  042516
 9787  065462  052125  040522  020114
 9788  065470  043101  042524  020122
 9789  065476  052101  042524  052116
 9790  065504  047511  020116  044502
 9791  065512  020124  051127  052111
 9792  065520  042524  000116
 9793
 9794  065524  051127  052111  020105  EM44:   .ASCIZ  /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
 9795  065532  052101  042524  052116
 9796  065540  047511  020116  044502
 9797  065546  020124  044504  020104
 9798  065554  047516  020124  042523
 9799  065562  020124  047520  052122
 9800  065570  051040  050505  042525
 9801  065576  052123     000
 9802
 9803  065601     103  047117  051124  EM45:   .ASCIZ  @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
 9804  065606  046117  042514  020122
 9805  065614  042523  042514  052103
 9806  065622  051440  044527  041524
 9807  065630  020110  047117  042040
 9808  065636  044522  042526  047040
 9809  065644  052117  044440  020116
 9810  065652  040447  041057  000047
 9811
 9812  065660  040503  023516  020124  EM46:   .ASCIZ  /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
 9813  065666  041501  042503  051523
 9814  065674  042040  044522  042526
 9815  065702  052040  051110  052517
 9816  065710  044107  042440  052111
 9817  065716  042510  020122  047520
 9818  065724  052122     000
 9819
 9820  065727     101  052124  020116  EM47:   .ASCIZ  /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
```

```
 9821  065734  044502  020124  047506
 9822  065742  020122  042523  055111
 9823  065750  047111  020107  047520
 9824  065756  052122  047040  052117
 9825  065764  041440  042514  051101
 9826  065772  042105  041040  020131
 9827  066000  040515  051523  052502
 9828  066006  020123  047111  052111
 9829  066014     000
 9830
 9831  066015     101  052124  020116  EM50:   .ASCIZ  /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
 9832  066022  044502  020124  047506
 9833  066030  020122  050117  047520
 9834  066036  044523  042524  050040
 9835  066044  051117  020124  046103
 9836  066052  040505  042522  020104
 9837  066060  054502  046440  051501
 9838  066066  041123  051525  044440
 9839  066074  044516  000124
 9840
 9841  066100  052101  047124  041040  EM51:   .ASCIZ  /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
 9842  066106  052111  041440  042514
 9843  066114  051101  042105  041040
 9844  066122  020131  040515  051523
 9845  066130  052502  020123  047111
 9846  066136  052111  020054  051104
 9847  066144  053111  020105  047111
 9848  066152  047040  052505  051124
 9849  066160  046101     000
 9850
 9851  066163     124  042510  040440  EM52:   .ASCIZ  /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
 9852  066170  052124  020116  044502
 9853  066176  020124  051511  051440
 9854  066204  052105  040440  052106
 9855  066212  051105  052040  046511
 9856  066220  047505  052125  053440
 9857  066226  052111  020110  047516
 9858  066234  051040  050505  042525
 9859  066242  052123  023040  023440
 9860  066250  051105  023522  051440
 9861  066256  052105     000
 9862
 9863  066261     103  047101  052047  EM53:   .ASCIZ  /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
 9864  066266  051040  040505  020104
 9865  066274  044124  020105  052101
 9866  066302  047124  041040  052111
 9867  066310  043040  047522  020115
 9868  066316  044124  020105  047447
 9869  066324  050120  051517  052111
 9870  066332  023505  050040  051117
 9871  066340  000124
 9872
 9873  066342  042522  042514  051501  EM54:   .ASCIZ  /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
 9874  066350  020105  047503  046515
 9875  066356  047101  020104  042522
 9876  066364  047503  047107  055111
```

```
 9877  066372  042105  053440  042510
 9878  066400  020116  051511  052523
 9879  066406  042105  041040  020131
 9880  066414  047516  026516  042523
 9881  066422  055111  047111  020107
 9882  066430  047520  052122     000
 9883
 9884  066435     124  046511  047505  EM55:   .ASCIZ  /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
 9885  066442  052125  047440  042516
 9886  066450  051455  047510  020124
 9887  066456  051511  046040  051505
 9888  066464  020123  044124  047101
 9889  066472  032440  030060  046440
 9890  066500  000123
 9891
 9892  066502  044122  030461  042040  EM56:   .ASCIZ  /RH11 DIDN'T RESPOND TO ADDRESSING/
 9893  066510  042111  023516  020124
 9894  066516  042522  050123  047117
 9895  066524  020104  047524  040440
 9896  066532  042104  042522  051523
 9897  066540  047111  000107
 9898
 9899
 9900
 9901
 9902
 9903  066544  042524  052123  021440  DH1:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
 9904  066552  020040  051105  020122
 9905  066560  041520  020040  047520
 9906  066566  052122  021440  020040
 9907  066574  042522  020107  042101
 9908  066602  020122  047503  052116
 9909  066610  047105  051524     000
 9910  066615     124  051505  020124  DH3:    .ASCIZ  /TEST #  ERR PC  REG ADR PORT A  PORT B/
 9911  066622  020043  042440  051122
 9912  066630  050040  020103  051040
 9913  066636  043505  040440  051104
 9914  066644  050040  051117  020124
 9915  066652  020101  050040  051117
 9916  066660  020124  000102
 9917  066664  020040  020040  020040  DH4:    .ASCII  /                SEIZE   ERROR/<CR><LF>
 9918  066672  020040  020040  020040
 9919  066700  020040  020040  042523
 9920  066706  055111  020105  020040
 9921  066714  051105  047522  006522
 9922  066722     012
 9923  066723     124  051505  020124          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
 9924  066730  020043  042440  051122
 9925  066736  050040  020103  050040
 9926  066744  051117  020124  020043
 9927  066752  050040  051117  020124
 9928  066760  020043  051040  043505
 9929  066766  040440  051104  043440
 9930  066774  047517  020104  020040
 9931  067002  041040  042101     000
 9932  067007     124  051505  020124  DH5:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR GOOD    BAD/
```

```
 9933  067014  020043  042440  051122
 9934  067022  050040  020103  050040
 9935  067030  051117  020124  020043
 9936  067036  051040  043505  040440
 9937  067044  051104  043440  047517
 9938  067052  020104  020040  041040
 9939  067060  042101     000
 9940  067063     040  020040  020040  DH7:    .ASCII  /                RELSNG  ERROR/<CR><LF>
 9941  067070  020040  020040  020040
 9942  067076  020040  020040  051040
 9943  067104  046105  047123  020107
 9944  067112  042440  051122  051117
 9945  067120  005015
 9946  067122  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
 9947  067130  020040  051105  020122
 9948  067136  041520  020040  047520
 9949  067144  052122  021440  020040
 9950  067152  047520  052122  021440
 9951  067160  020040  042522  020107
 9952  067166  042101  020122  047507
 9953  067174  042117  020040  020040
 9954  067202  040502  000104
 9955  067206  042524  052123  021440  DH11:   .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
 9956  067214  020040  051105  020122
 9957  067222  041520  020040  047520
 9958  067230  052122  021440  020040
 9959  067236  042522  020107  042101
 9960  067244  020122  047503  052116
 9961  067252  047105  051524     000
 9962  067257     040  020040  020040  DH13:   .ASCII  /                SEIZE   ERROR/<CR><LF>
 9963  067264  020040  020040  020040
 9964  067272  020040  020040  051440
 9965  067300  044505  042532  020040
 9966  067306  042440  051122  051117
 9967  067314  005015
 9968  067316  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR CONTENTS/
 9969  067324  020040  051105  020122
 9970  067332  041520  020040  047520
 9971  067340  052122  021440  020040
 9972  067346  047520  052122  021440
 9973  067354  020040  042522  020107
 9974  067362  042101  020122  047503
 9975  067370  052116  047105  051524
 9976  067376     000
 9977  067377     040  020040  020040  DH22:   .ASCII  /                RELSNG  SEIZE/<CR><LF>
 9978  067404  020040  020040  020040
 9979  067412  020040  020040  051040
 9980  067420  046105  047123  020107
 9981  067426  051440  044505  042532
 9982  067434  005015
 9983  067436  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
 9984  067444  020040  051105  020122
 9985  067452  041520  020040  047520
 9986  067460  052122  021440  020040
 9987  067466  047520  052122  021440
 9988  067474     000
```

```
 9989  067475     040  020040  020040  DH23:   .ASCII  /               SEIZE/<CR><LF>
 9990  067502  020040  020040  020040
 9991  067510  020040  020040  051440
 9992  067516  044505  042532  005015
 9993  067524  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
 9994  067532  020040  051105  020122
 9995  067540  041520  020040  047520
 9996  067546  052122  021440  020040
 9997  067554  042522  020107  042101
 9998  067562  020122  047503  052116
 9999  067570  047105  051524     000
10000  067575     040  020040  020040  DH26:   .ASCII  /               RELSNG/<CR><LF>
10001  067602  020040  020040  020040
10002  067610  020040  020040  051040
10003  067616  046105  047123  006507
10004  067624     012
10005  067625     124  051505  020124          .ASCIZ  /TEST #  ERR PC  PORT #/
10006  067632  020043  042440  051122
10007  067640  050040  020103  050040
10008  067646  051117  020124  000043
10009  067654  020040  020040  020040  DH31:   .ASCII  /               RELSNG  RQSTNG/<CR><LF>
10010  067662  020040  020040  020040
10011  067670  020040  020040  042522
10012  067676  051514  043516  020040
10013  067704  050522  052123  043516
10014  067712  005015
10015  067714  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10016  067722  020040  051105  020122
10017  067730  041520  020040  047520
10018  067736  052122  021440  020040
10019  067744  047520  052122  021440
10020  067752     000
10021  067753     124  051505  020124  DH36:   .ASCIZ  /TEST #  ERR PC  PORT #/
10022  067760  020043  042440  051122
10023  067766  050040  020103  050040
10024  067774  051117  020124  000043
10025  070002  042524  052123  021440  DH42:   .ASCIZ  /TEST #  ERR PC/
10026  070010  020040  051105  020122
10027  070016  041520     000
10028  070021     040  020040  020040  DH44:   .ASCII  /               RELSNG  ERROR/<CR><LF>
10029  070026  020040  020040  020040
10030  070034  020040  020040  051040
10031  070042  046105  047123  020107
10032  070050  042440  051122  051117
10033  070056  005015
10034  070060  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10035  070066  020040  051105  020122
10036  070074  041520  020040  047520
10037  070102  052122  021440  020040
10038  070110  047520  052122  021440
10039  070116     000
10040  070117     040  020040  020040  DH46:   .ASCII  /               PORT A  PORT B/<CR><LF>
10041  070124  020040  020040  020040
10042  070132  020040  020040  050040
10043  070140  051117  020124  020101
10044  070146  050040  051117  020124
```

```
10045  070154  006502     012
10046  070157     124  051505  020124          .ASCIZ  /TEST #  ERR PC  RPDS1   RPDS1/
10047  070164  020043  042440  051122
10048  070172  050040  020103  051040
10049  070200  042120  030523  020040
10050  070206  051040  042120  030523
10051  070214     000
10052  070215     124  051505  020124  DH55:   .ASCIZ  /TEST #  ERR PC  PORT #  TIMEOUT VALUE (IN MS)/
10053  070222  020043  042440  051122
10054  070230  050040  020103  050040
10055  070236  051117  020124  020043
10056  070244  052040  046511  047505
10057  070252  052125  053040  046101
10058  070260  042525  024040  047111
10059  070266  046440  024523     000
10060  070273     044  050122  042101  DH56:   .ASCIZ  /$RPADR/
10061  070300  000122
10062
10063                                  .EVEN
10064
10065  070302  001242  001116  001234  DT1:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$BDDAT,0
10066  070310  001122  001126  000000
10067  070316  001242  001116  001122  DT3:    .WORD   TSTNUM,$ERRPC,$BDADR,$GDDAT,$BDDAT,0
10068  070324  001124  001126  000000
10069  070332  001242  001116  001234  DT5:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10070  070340  001122  001124  001126
10071  070346  000000
10072  070350  001242  001116  001240  DT6:    .WORD   TSTNUM,$ERRPC,OPPRT,$BDADR,$BDDAT,0
10073  070356  001122  001126  000000
10074  070364  001242  001116  001236  DT7:    .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10075  070372  001234  001122  001124
10076  070400  001126  000000
10077  070404  001242  001116  001236  DT13:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$BDDAT,0
10078  070412  001234  001122  001126
10079  070420  000000
10080  070422  001242  001116  001236  DT22:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,0
10081  070430  001234  000000
10082  070434  001242  001116  001236  DT23:   .WORD   TSTNUM,$ERRPC,SEIZPT,$BDADR,$BDDAT,0
10083  070442  001122  001126  000000
10084  070450  001242  001116  001236  DT31:   .WORD   TSTNUM,$ERRPC,SEIZPT,OPPRT,0
10085  070456  001240  000000
10086  070462  001242  001116  001236  DT36:   .WORD   TSTNUM,$ERRPC,SEIZPT,0
10087  070470  000000
10088  070472  001242  001116  001234  DT37:   .WORD   TSTNUM,$ERRPC,PTNBR,0
10089  070500  000000
10090  070502  001242  001116  000000  DT42:   .WORD   TSTNUM,$ERRPC,0
10091  070510  001242  001116  001170  DT46:   .WORD   TSTNUM,$ERRPC,$TMP2,$TMP3,0
10092  070516  001172  000000
10093  070522  001242  001116  001240  DT54:   .WORD   TSTNUM,$ERRPC,OPPRT,SEIZPT,0
10094  070530  001236  000000
10095  070534  001242  001116  001236  DT55:   .WORD   TSTNUM,$ERRPC,SEIZPT,TIME,0
10096  070542  001262  000000
10097  070546  001300  000000          DT56:   .WORD   $RPADR,0
10098
10099  070552     000     000     000  DF1:    .BYTE   0,0,0,0,0
10100  070555     000     000
```

```
10101  070557  000      000      000  DF5:    .BYTE   0,0,0,0,0,0
10102  070562  000      000      000
10103  070565  000      000      000  DF7:    .BYTE   0,0,0,0,0,0,0
10104  070570  000      000      000
10105  070573  000
10106  070574  000      000      000  DF31:   .BYTE   0,0,0,0
10107  070577  000
10108  070600  000      000      000  DF36:   .BYTE   0,0,0
10109  070603  000      000           DF42:   .BYTE   0,0
10110  070605  000      000      000  DF55:   .BYTE   0,0,0,1
10111  070610  001
10112  070611  000                   DF56:   .BYTE   0
10113
10114                                         .EVEN
10115
10116
10117                                 ;;*****************************************************************
10118
10119                                 .SBTTL  CONSTANTS, TABLES, ETC
10120
10121                                 ;;*****************************************************************
10122
10123                                 ;TABLE OF TEST STARTING ADDRESSES
10124
10125  070612  003070                 TSTADR: .WORD   TST1             ;STARTING ADDRESS OF TEST 1
10126  070614  004470                         .WORD   TST2             ;STARTING ADDRESS OF TEST 2
10127  070616  006070                         .WORD   TST3             ;STARTING ADDRESS OF TEST 3
10128  070620  007470                         .WORD   TST4             ;STARTING ADDRESS OF TEST 4
10129  070622  010626                         .WORD   TST5             ;STARTING ADDRESS OF TEST 5
10130  070624  011764                         .WORD   TST6             ;STARTING ADDRESS OF TEST 6
10131  070626  012754                         .WORD   TST7             ;STARTING ADDRESS OF TEST 7
10132  070630  013744                         .WORD   TST10            ;STARTING ADDRESS OF TEST 10
10133  070632  014366                         .WORD   TST11            ;STARTING ADDRESS OF TEST 11
10134  070634  015010                         .WORD   TST12            ;STARTING ADDRESS OF TEST 12
10135  070636  016366                         .WORD   TST13            ;STARTING ADDRESS OF TEST 13
10136  070640  017744                         .WORD   TST14            ;STARTING ADDRESS OF TEST 14
10137  070642  021246                         .WORD   TST15            ;STARTING ADDRESS OF TEST 15
10138  070644  022550                         .WORD   TST16            ;STARTING ADDRESS OF TEST 16
10139  070646  024010                         .WORD   TST17            ;STARTING ADDRESS OF TEST 17
10140  070650  024676                         .WORD   TST20            ;STARTING ADDRESS OF TEST 20
10141  070652  025564                         .WORD   TST21            ;STARTING ADDRESS OF TEST 21
10142  070654  026744                         .WORD   TST22            ;STARTING ADDRESS OF TEST 22
10143  070656  030124                         .WORD   TST23            ;STARTING ADDRESS OF TEST 23
10144  070660  031134                         .WORD   TST24            ;STARTING ADDRESS OF TEST 24
10145  070662  032144                         .WORD   TST25            ;STARTING ADDRESS OF TEST 25
10146  070664  033352                         .WORD   TST26            ;STARTING ADDRESS OF TEST 26
10147  070666  034560                         .WORD   TST27            ;STARTING ADDRESS OF TEST 27
10148  070670  036554                         .WORD   TST30            ;STARTING ADDRESS OF TEST 30
10149  070672  037250                         .WORD   TST31            ;STARTING ADDRESS OF TEST 31
10150  070674  040510                         .WORD   TST32            ;STARTING ADDRESS OF TEST 32
10151  070676  041750                         .WORD   TST33            ;STARTING ADDRESS OF TEST 33
10152  070700  043040                         .WORD   TST34            ;STARTING ADDRESS OF TEST 34
10153  070702  044130                         .WORD   TST35            ;STARTING ADDRESS OF TEST 35
10154  070704  044744                         .WORD   TST36            ;STARTING ADDRESS OF TEST 36
10155  070706  045560                         .WORD   TST37            ;STARTING ADDRESS OF TEST 37
10156  070710  046542                         .WORD   TST40            ;STARTING ADDRESS OF TEST 40
```

```
10157  070712  047524                          .WORD   TST41           ;STARTING ADDRESS OF TEST 41
10158  070714  051132                          .WORD   TST42           ;STARTING ADDRESS OF TEST 42
10159  070716  052540                          .WORD   TST43           ;STARTING ADDRESS OF TEST 43
10160  070720  053422                          .WORD   TST44           ;STARTING ADDRESS OF TEST 44
10161  070722  054310                          .WORD   TST45           ;STARTING ADDRESS OF TEST 45
10162  070724  055256                          .WORD   TST46           ;STARTING ADDRESS OF TEST 46
10163
10164                                  ;ATTENTION BIT TABLE
10165
10166  070726     001                  ATABIT: .BYTE   1               ;ATTENTION BIT FOR DRIVE 0
10167  070727     002                          .BYTE   2               ;ATTENTION BIT FOR DRIVE 1
10168  070730     004                          .BYTE   4               ;ATTENTION BIT FOR DRIVE 2
10169  070731     010                          .BYTE   10              ;ATTENTION BIT FOR DRIVE 3
10170  070732     020                          .BYTE   20              ;ATTENTION BIT FOR DRIVE 4
10171  070733     040                          .BYTE   40              ;ATTENTION BIT FOR DRIVE 5
10172  070734     100                          .BYTE   100             ;ATTENTION BIT FOR DRIVE 6
10173  070735     200                          .BYTE   200             ;ATTENTION BIT FOR DRIVE 7
10174
10175  070736  000046                  MAXTN:  .WORD   $TN-1           ;MAXIMUM TEST NUMBER
10176
10177           000001                          .END
```

```
ABS    = 000200          1384#
ACL    = 000040          1419#   1428#
ACU    = 100000          1373#
ADDRIS   062374          2069    9465#
ADRERR   062147          1996    9437#
AOE    = 001000          1290#
ASR1     001232          1560#   2021*   5904    6178    7617    7831    8024    8025    8138    8139
ATA    = 100000          1277#   2419    2423    2444    2450    2656    2660    2681    2687    2909    2913    2934    2940
                         3088    3092    3113    3119    3183    3187    3208    3214    3230    3234    3247    3251    3328
                         3332    3353    3359    3375    3379    3392    3396    3460    3464    3485    3491    3540    3544
                         3565    3571    3743    3747    3770    3776    3947    3951    3974    3980    4043    4045    4049
                         4063    4065    4069    4102    4106    4129    4133    4156    4162    4177    4179    4183    4244
                         4246    4250    4264    4266    4270    4303    4307    4330    4334    4357    4363    4378    4380
                         4384    4443    4445    4449    4463    4465    4469    4489    4493    4542    4544    4548    4559
                         4561    4565    4657    4661    4779    4783    4889    4916    4920    4931    4933    4937    4960
                         4964    5069    5096    5100    5111    5113    5117    5140    5144    5290    5294    5439    5443
                         5594    5629    5633    5783    5818    5822    5959    5994    5998    6065    6100    6104    6195
                         6199    6344    6346    6350    6368    6372    6395    6401    6418    6422    6546    6548    6552
                         6570    6574    6597    6603    6620    6624    6770    6774    6939    6943    7068    7072    7205
                         7209    7315    7317    7321    7338    7342    7365    7369    7463    7465    7469    7486    7490
                         7513    7517    7636    7671    7675    7850    7885    7889    8055    8059    8082    8088    8169
                         8173    8196    8202    8343    8347    8508    8512
ATABIT   070726          2021   10166#
ATO    = 000001          1310#
AT1    = 000002          1311#
AT2    = 000004          1312#
AT3    = 000010          1313#
AT4    = 000020          1314#
AT5    = 000040          1315#
AT6    = 000100          1316#
AT7    = 000200          1317#
A16    = 000400          1208#
A17    = 001000          1209#
BADNO    062334          2054    9459#
BAI    = 000010          1226#
BIT0   = 000001          1182#
BIT00  = 000001          1172#   1182
BIT01  = 000002          1171#   1181
BIT02  = 000004          1170#   1180
BIT03  = 000010          1169#   1179
BIT04  = 000020          1168#   1178
BIT05  = 000040          1167#   1177
BIT06  = 000100          1166#   1176
BIT07  = 000200          1165#   1175
BIT08  = 000400          1164#   1174
BIT09  = 001000          1163#   1173    8694
BIT1   = 000002          1181#
BIT10  = 002000          1162#   8738
BIT11  = 004000          1161#   8701
BIT12  = 010000          1160#
BIT13  = 020000          1159#   8745
BIT14  = 040000          1158#   8676
BIT15  = 100000          1157#
BIT2   = 000004          1180#
BIT3   = 000010          1179#
BIT4   = 000020          1178#
```

```
BIT5   = 000040          1177#
BIT6   = 000100          1176#
BIT7   = 000200          1175#
BIT8   = 000400          1174#
BIT9   = 001000          1173#
BPTVEC= 000014           1189#
CHANGE  002754           1990    2066#
CHGADR  001276           1578#   1937*   1939*   2066    2068*
CKCLK   056444           2026    2039    8610#
CKCLK1  056514           8610    8619#
CKCLK2  056556           8620    8627#
CKCLK3  056566           8618    8626    8629#
CKERR   001244           1565#   2129*   2142*   2144    2150*   2163*   2165    2174*   2187*   2191*   2204*   2212*   2225*
                         2227*   2240*   2244*   2257*   2259*   2272*   2815*   2823*   2827*   2840*   2842*   2855*   2867*
                         2880*   2994*   3002*   3006*   3019*   3021*   3034*   3046*   3059*   3224*   3237*   3241*   3254*
                         3369*   3382*   3386*   3399*   3652*   3665*   3669*   3682*   3695*   3708*   3712*   3725*   3856*
                         3869*   3873*   3886*   3899*   3912*   3916*   3929*   4039*   4052*   4054    4059*   4072*   4074
                         4096*   4109*   4173*   4186*   4240*   4253*   4255    4260*   4273*   4275    4297*   4310*   4374*
                         4387*   4439*   4452*   4454    4459*   4472*   4474    4538*   4551*   4555*   4568*   4910*   4923*
                         4927*   4940*   5090*   5103*   5107*   5120*   5255*   5268*   5404*   5417*   5562*   5570*   5579
                         5751*   5759*   5768    5932*   5945*   6038*   6051*   6307*   6320*   6326*   6334*   6340*   6353*
                         6412*   6425*   6509*   6522*   6528*   6536*   6542*   6555*   6614*   6627*   6695*   6708*   6710
                         6725*   6733*   6737*   6745*   6864*   6877*   6879    6894*   6902*   6906*   6914*   7311*   7324*
                         7332*   7345*   7459*   7472*   7480*   7493*   7713*   7721*   7730*   7738*   7747*   7755*   7927*
                         7935*   7944*   7952*   7961*   7969*   8292*   8300*   8302    8312*   8320*   8322    8457*   8465*
                         8467    8477*   8485*   8487
CKSWR  = 104407          8675    8733    8753    9406#
CLOCK   056576           8614    8623    8634#
CLR    = 000040          1228#   2146    2167    3688    3892    4090    4291    4531    6713    6882
CR     = 000015          1097#   1550#   8876    8886    9419    9423    9432    9437    9441    9445    9449    9459    9463
                         9465    9917    9940    9962    9977    9989    10000   10009   10028   10040
CRLF   = 000200          1098#   8847    8886
CSF    = 000002          1360#   1378#
CSU    = 000010          1362#   1380#
DCK    = 100000          1296#
DCL    = 000100          1420#   1429#
DCU    = 000001          1426#
DDISP  = 177570          1104#   1526    1966
DE1    = 000040          1267#
DFF20  = 000002          1263#
DF1     070552           1610    1617    1624    1666    1736    1743    1829    1836    1864    1878    1906    10099#
DF31    070574           1729    1759    1780    1857    1871    1913    10106#
DF36    070600           1673    1715    1722    1751    1773    1794    1801    1808    1815    1822    10108#
DF42    070603           1843    1850    10109#
DF5     070557           1638    1645    1659    1680    1687    1787    1885    1892    1899    10101#
DF55    070605           1920    10110#
DF56    070611           1927    10112#
DF7     070565           1631    1652    1694    1701    1708    1766    10103#
DH1     066544           1608    1615    1827    1862    9903#
DH11    067206           1664    9955#
DH13    067257           1643    1678    1685    1883    1897    9962#
DH22    067377           1727    1757    1911    9977#
DH23    067475           1734    1741    1834    1876    1904    9989#
DH26    067575          10000#
DH3     066615           1622    9910#
DH31    067654           1778   10009#
```

F15


```
DH36     067753          1671    1713    1720    1749    1771    1792    1799    1806    1813    1820    10021#
DH4      066664          1629    9917#
DH42     070002          1841    1848    10025#
DH44     070021          1855    10028#
DH46     070117          1869    10040#
DH5      067007          1636    1657    1785    1890    9932#
DH55     070215          1918    10052#
DH56     070273          1925    10060#
DH7      067063          1650    1692    1699    1706    1764    9940#
DIGB   = 000004          1264#
DISPLA   001142          1526#   1966*   1974*   8715*   8737*
DISPRE   000174          1475#   1974
DLT    = 100000          1238#
DL64   = 000020          1266#
DMD    = 000001          1300#
DPR    = 000400          1270#   2231    2263    2353    2415    2590    2652    2831    2871    2905    3010    3050    3084
                         3179    3324    3456    3536    3632    3673    3716    3739    3836    3877    3920    3943    4125
                         4326    4630    4653    4752    4775    4889    4956    5069    5136    5240    5286    5389    5435
                         5542    5594    5625    5731    5783    5814    5936    5959    5990    6042    6065    6096    6191
                         6766    6935    7064    7201    7361    7509    7636    7667    7850    7881    8051    8165    8339
                         8504
DRQ    = 004000          1333#
DRY    = 000200          1269#   2231    2263    2353    2415    2590    2652    2831    2871    2905    3010    3050    3084
                         3179    3324    3456    3536    3632    3673    3716    3739    3836    3877    3920    3943    4125
                         4326    4630    4653    4752    4775    4889    4956    5069    5136    5240    5286    5389    5435
                         5542    5594    5625    5731    5783    5814    5936    5959    5990    6042    6065    6096    6191
                         6766    6935    7064    7201    7305    7361    7453    7509    7636    7667    7850    7881    8051
                         8165    8339    8504
DSWR   = 177570          1103#   1525    1965
DTE    = 010000          1293#
DTSY   = 000200          1306#
DT00   = 000001          1324#
DT01   = 000002          1325#
DT02   = 000004          1326#
DT03   = 000010          1327#
DT04   = 000020          1328#
DT05   = 000040          1329#
DT06   = 000100          1330#
DT07   = 000200          1331#
DT08   = 000400          1332#
DT1      070302          1609    1616    1665    1863    1905    10065#
DT13     070404          1644    1679    1686    1884    1898    10077#
DT22     070422          1728    1758    10080#
DT23     070434          1735    1742    1828    1835    1877    10082#
DT3      070316          1623    10067#
DT31     070450          1779    1856    10084#
DT36     070462          1750    1772    1793    1800    1814    10086#
DT37     070472          1672    1714    1721    1807    1821    10088#
DT42     070502          1842    1849    10090#
DT46     070510          1870    10091#
DT5      070332          1637    1658    1786    1891    10069#
DT54     070522          1912    10093#
DT55     070534          1919    10095#
DT56     070546          1926    10097#
DT6      070350          10072#
DT7      070364          1630    1651    1693    1700    1707    1765    10074#
```

```
DVA    = 004000          1258#
ECH    = 000100          1287#
ECI    = 004000          1401#
EMTVEC= 000030           1192#   1951*   1952*
EM1     062506           1607    9485#
EM10    063113           1656    9540#
EM11    063143           1663    9546#
EM12    063227           1670    9556#
EM13    063277           1677    9564#
EM14    063364           1684    9574#
EM15    063427           1691    9581#
EM16    063506           1698    9590#
EM17    063561           1705    9599#
EM2     062527           1614    9489#
EM20    063640           1712    9608#
EM21    063720           1719    9617#
EM22    063773           1726    9626#
EM23    064060           1733    9636#
EM24    064126           1740    9644#
EM25    064205           1748    9653#
EM26    064250           1756    9660#
EM27    064335           1763    9670#
EM3     062551           1621    9494#
EM30    064413           1770    9679#
EM31    064510           1777    9691#
EM32    064565           1784    9700#
EM33    064636           1791    9708#
EM34    064740           1798    9720#
EM35    065043           1805    9733#
EM36    065122           1812    9742#
EM37    065174           1819    9750#
EM4     062633           1628    9504#
EM40    065242           1826    9758#
EM41    065317           1833    9767#
EM42    065361           1840    9774#
EM43    065447           1847    9785#
EM44    065524           1854    9794#
EM45    065601           1861    9803#
EM46    065660           1868    9812#
EM47    065727           1875    9820#
EM5     062664           1635    9510#
EM50    066015           1882    9831#
EM51    066100           1889    9841#
EM52    066163           1896    9851#
EM53    066261           1903    9863#
EM54    066342           1910    9873#
EM55    066435           1917    9884#
EM56    066502           1924    9892#
EM6     062732           1642    9518#
EM7     063032           1649    9530#
ENTERA  062117           1991    9432#
ERR    = 040000          1276#   6311    6513
ERRVEC= 000004           1185#   1963    1964*   1975*   8610*   8611*   8620*   8629*   8681    8682*   8684*   8687*
EXEC    002602           2027    2034#   2115    2313    2550    2788    2967    3143    3288    3431    3511    3602    3806
                         4011    4212    4411    4589    4711    4841    5021    5199    5348    5501    5690    5876    6155
                         6252    6454    6663    6832    6998    7135    7275    7423    7581    7795    7998    8112    8242
```

```
                        8407    8560
EXT1   = 000001         1340*
EXT10  = 000010         1343*
EXT2   = 000002         1341*
EXT20  = 000020         1344*
EXT4   = 000004         1342*
EXT40  = 000040         1345*
FEN    = 000200         1366*
FER    = 000020         1285*
FMT22  = 010000         1402*   2862    3041
F1     = 000002         1253*
F2     = 000004         1254*
F3     = 000010         1255*
F4     = 000020         1256*
F5     = 000040         1257*
GNS    = ****** U       1474    8573    8580    9398    9399    9400    9401    9402    9404    9406    9407    9408    9409
                        9410    9411
GO     = 000001         1252*
GRV    = 000010         1265*
GTSWR  = 104406         1986    9404*
HCE    = 000200         1288*
HCI    = 002000         1400*
HCRC   = 000400         1289*
HT     = 000011         1095*   8845    8886
IAE    = 002000         1291*
IE     = 000100         1206*
ILF    = 000001         1281*
ILR    = 000002         1282*
IOTVEC= 000020          1190*   1949*   1950*
IR     = 000100         1229*
IXE    = 004000         1370*   1388*
KYBCTL  001274          1577*   2042*   2059*   2112    2116*   2310    2314*   2547    2551*   2785    2789*   2964    2968*
                        3140    3144*   3285    3289*   3428    3432*   3508    3512*   3599    3603*   3803    3807*   4008
                        4012*   4209    4213*   4408    4412*   4586    4590*   4708    4712*   4838    4842*   5018    5022*
                        5196    5200*   5345    5349*   5498    5502*   5687    5691*   5873    5877*   6152    6156*   6249
                        6253*   6451    6455*   6660    6664*   6829    6833*   6995    6999*   7132    7136*   7272    7276*
                        7420    7424*   7578    7582*   7792    7796*   7995    7999*   8109    8113*   8239    8243*   8404
                        8408*   8558
LF     = 000012         1096*   1551*   8880    8886    9419    9423    9432    9437    9441    9445    9449    9456    9459
                        9463    9465    9473    9917    9940    9962    9977    9989    10000   10009   10028   10040
LST    = 002000         1272*
MAXTN   070735          2052    10175*
MCLK   = 000002         1301*
MCPE   = 020000         1211*   5574    5763    7725    7742    7759    7939    7956    7973
MHS    = 001000         1368*   1386*
MINX   = 000004         1302*
MOH    = 020000         1334*
MOL    = 010000         1274*   2231    2263    2353    2415    2590    2652    2831    2871    2905    3010    3050    3084
                        3179    3324    3456    3536    3632    3673    3716    3739    3836    3877    3920    3943    4125
                        4326    4630    4653    4752    4775    4889    4956    5069    5136    5240    5286    5389    5435
                        5542    5594    5625    5731    5783    5814    5936    5959    5990    6042    6065    6096    6191
                        6766    6935    7064    7201    7361    7509    7636    7667    7850    7881    8051    8165    8339
                        8504
MPE    = 000400         1231*
MRD    = 000020         1304*
MSE    = 000020         1363*
```

```
MSTCK = 000010           1303#
MWR   = 000040           1305#
MXF   = 001000           1232#
NBA   = 100000           1336#
NED   = 010000           1235#  2135   2139   2156   2160
NEM   = 004000           1234#
NHS   = 002000           1369#  1387#
NOATA = 000001           1931#  2444   2450   2681   2687   2934   2940   3113   3119   3208   3214   3353   3359
                         3485   3491   3565   3571   3770   3776   3974   3980   4156   4162   4357   4363   4391#
                         4516   4521   4684   4689   4806   4811   4876#  4943#  4987   4992   5056#  5123#  5167
                         5172   5317   5322   5466   5471   5656   5661   5845   5850   6021   6026   6127   6132
                         6222   6227   6358#  6395   6401   6406#  6560#  6597   6603   6608#  6797   6802   6966
                         6971   7095   7100   7232   7237   7392   7397   7540   7545   7698   7703   7912   7917
                         8038#  8082   8088   8093#  8152#  8196   8202   8207#  8370   8375   8535   8540
NOCLOC  062245           2028   9449#
NOSEIZ  001246           1566#
NTRH11  062454           2073   9473#
OCYL  = 100000           1422#  1432#
OFREV = 000200           1399#
OF100 = 000004           1395#
OF200 = 000010           1396#
OF25  = 000001           1393#
OF400 = 000020           1397#
OF50  = 000002           1394#
OF800 = 000040           1398#
OPE   = 020000           1430#
OPI   = 020000           1294#
OPPRT   001240           1563#  2340#  2577#  3164#  3309#  3619#  3823#  4085#  4286#  4617#  4739#  4867#  5047#
                         5227#  5376#  5529#  5718#  5948#  6054#  6277#  6479#  6688#  6857#  7024#  7161#  7605#
                         7819#  8268#  8433#  10072  10084  10093
OR    = 000200           1230#
PAR   = 000010           1284#
PAT   = 000020           1227#
PGE   = 002000           1233#
PGM   = 001000           1271#  2216   2218   2222   2248   2250   2254   2353   2415   2590   2652   2831   2871
                         2905   3010   3050   3084   3179   3324   3456   3536   3632   3673   3716   3739   3836
                         3877   3920   3943   4125   4326   4630   4653   4752   4775   4889   4956   5069   5136
                         5240   5286   5389   5435   5542   5594   5625   5731   5783   5814   5936   5959   5990
                         6042   6065   6096   6191   6766   6935   7064   7201   7361   7509   7636   7667   7850
                         7881   8051   8165   8339   8504
PIP   = 020000           1275#
PIRQ  = 177772           1102#
PIRQVE= 000240           1196#
PLU   = 020000           1372#  1389#
PORTA   001224           1557#  1993#  1994   1998   2001   2005   2017   2020   2126   2127   2172   2173   2210
                         2211   2278   2335   2336   2350   2351   2416   2435   2437   2443   2575   2576   2577
                         2603   2604   2653   2672   2674   2680   2697   2795   2796   2803   2825   2826   2906
                         2925   2927   2933   2992   2993   3061   3062   3085   3104   3106   3112   3161   3162
                         3170   3171   3180   3199   3201   3207   3222   3223   3309   3325   3344   3346   3352
                         3384   3385   3408   3409   3438   3439   3440   3457   3476   3478   3484   3537   3556
                         3558   3564   3614   3615   3629   3630   3667   3668   3710   3711   3730   3731   3740
                         3759   3761   3769   3821   3822   3823   3854   3855   3897   3898   3944   3963   3965
                         3973   4022   4030   4037   4038   4082   4083   4091   4116   4117   4126   4145   4147
                         4155   4223   4231   4258   4259   4286   4327   4346   4348   4356   4372   4373   4422
                         4430   4437   4438   4486   4505   4507   4515   4536   4537   4599   4612   4613   4627
                         4628   4644   4645   4654   4673   4675   4683   4721   4737   4738   4739   4776   4795
```

```
                    4797  4805  4851  4867  4868  4869  4892  4893  4903  4925  4926  4947  4948
                    4957  4976  4978  4986  5031  5044  5045  5062  5063  5075  5076  5088  5089
                    5137  5156  5158  5166  5209  5225  5226  5227  5249  5250  5287  5306  5308
                    5316  5358  5371  5372  5386  5387  5426  5427  5436  5455  5457  5465  5511
                    5527  5528  5529  5555  5556  5597  5598  5608  5616  5617  5626  5645  5647
                    5655  5700  5713  5714  5728  5729  5776  5777  5789  5790  5815  5834  5836
                    5844  5886  5899  5905  5906  5948  5962  5963  5973  5981  5982  5991  6010
                    6012  6020  6036  6037  6053  6058  6059  6071  6072  6097  6116  6118  6126
                    6165  6192  6211  6213  6221  6262  6274  6275  6301  6302  6365  6384  6386
                    6394  6430  6464  6479  6491  6492  6567  6586  6588  6596  6612  6613  6673
                    6685  6686  6714  6715  6735  6736  6757  6758  6767  6786  6788  6796  6842
                    6857  6892  6893  6936  6955  6957  6965  7008  7021  7022  7065  7084  7086
                    7094  7145  7161  7185  7186  7202  7221  7223  7231  7285  7293  7294  7295
                    7352  7353  7362  7381  7383  7391  7433  7474  7475  7510  7529  7531  7539
                    7591  7602  7603  7612  7613  7623  7624  7629  7630  7642  7643  7668  7687
                    7689  7697  7805  7819  7823  7824  7832  7833  7853  7854  7864  7872  7873
                    7882  7901  7903  7911  8008  8016  8052  8071  8073  8081  8122  8133  8142
                    8143  8144  8156  8157  8166  8185  8187  8195  8252  8268  8269  8270  8311
                    8330  8331  8340  8359  8361  8369  8417  8430  8431  8441  8442  8505  8524
                    8526  8534
PORTAI  062171       2004  9441#
PORTB   001226      1558#  1998*  1999*  2000*  2003*  2011  2147  2148  2189  2190  2242  2243  2280
                    2338  2339  2340  2366  2367  2420  2431  2432  2449  2460  2572  2573  2587
                    2588  2657  2668  2669  2686  2813  2814  2882  2883  2910  2921  2922  2939
                    2974  2975  2982  3004  3005  3089  3100  3101  3118  3164  3184  3195  3196
                    3213  3239  3240  3263  3264  3306  3307  3315  3316  3329  3340  3341  3358
                    3367  3368  3461  3472  3473  3490  3518  3519  3520  3541  3552  3553  3570
                    3617  3618  3619  3650  3651  3693  3694  3744  3755  3756  3775  3818  3819
                    3833  3834  3871  3872  3914  3915  3934  3935  3948  3959  3960  3979  4025
                    4057  4058  4085  4130  4141  4142  4161  4171  4172  4226  4238  4239  4283
                    4284  4292  4317  4318  4331  4342  4343  4362  4425  4457  4458  4490  4501
                    4502  4520  4553  4554  4603  4615  4616  4617  4658  4669  4670  4688  4725
                    4734  4735  4749  4750  4766  4767  4780  4791  4792  4810  4855  4864  4865
                    4882  4883  4895  4896  4908  4909  4961  4972  4973  4991  5035  5047  5048
                    5049  5072  5073  5083  5105  5106  5127  5128  5141  5152  5153  5171  5213
                    5222  5223  5237  5238  5277  5278  5291  5302  5303  5321  5362  5374  5375
                    5376  5398  5399  5440  5451  5452  5470  5515  5524  5525  5539  5540  5587
                    5588  5600  5601  5630  5641  5642  5660  5704  5716  5717  5718  5744  5745
                    5786  5787  5797  5805  5806  5819  5830  5831  5849  5890  5913  5914  5920
                    5921  5947  5952  5953  5965  5966  5995  6006  6007  6025  6054  6068  6069
                    6079  6087  6088  6101  6112  6113  6131  6169  6196  6207  6208  6226  6266
                    6277  6289  6290  6369  6380  6381  6400  6410  6411  6468  6476  6477  6503
                    6504  6571  6582  6583  6602  6632  6677  6688  6723  6724  6771  6782  6783
                    6801  6846  6854  6855  6883  6884  6904  6905  6926  6927  6940  6951  6952
                    6970  7012  7024  7048  7049  7069  7080  7081  7099  7149  7158  7159  7206
                    7217  7218  7236  7289  7326  7327  7366  7377  7378  7396  7437  7441  7442
                    7443  7500  7501  7514  7525  7526  7544  7595  7605  7609  7610  7618  7619
                    7639  7640  7650  7658  7659  7672  7683  7684  7702  7809  7816  7817  7826
                    7827  7837  7838  7843  7844  7856  7857  7886  7897  7898  7916  8012  8019
                    8028  8029  8030  8042  8043  8056  8067  8068  8087  8126  8130  8170  8181
                    8182  8201  8256  8265  8266  8276  8277  8344  8355  8356  8374  8421  8433
                    8434  8435  8476  8495  8496  8509  8520  8521  8539
PORTBI  062217       2010  9445#
PORTC   001230      1559#  2017*  2018*  2019*  5898  6173
PRE   = 000020      1418#
PRO   = 000000      1119#
```

```
PR1     = 000040            1120#
PR2     = 000100            1121#
PR3     = 000140            1122#
PR4     = 000200            1123#
PR5     = 000240            1124#
PR6     = 000300            1125#
PR7     = 000340            1126#
PS      = 177776            1099#  1100   2035*  2320*  2391*  2408*  2557*  2628*  2645*
PSEL    = 002000            1210#
PSU     = 000001            1415#
PSW     = 177776            1100#
PTNBR   = 001234            1561#  2127*  2148*  2173*  2190*  2211*  2243*  2339*  2351*  2367*  2431*  2435*  2443*
                            2449*  2460*  2576*  2588*  2604*  2668*  2672*  2680*  2686*  2697*  2796*  2814*  2826*
                            2883*  2921*  2925*  2933*  2939*  2975*  2993*  3005*  3062*  3100*  3104*  3112*  3118*
                            3171*  3195*  3199*  3207*  3213*  3223*  3240*  3264*  3316*  3340*  3344*  3352*  3358*
                            3368*  3385*  3409*  3439*  3472*  3476*  3484*  3490*  3519*  3552*  3556*  3564*  3570*
                            3618*  3630*  3651*  3668*  3694*  3711*  3731*  3755*  3759*  3769*  3775*  3822*  3834*
                            3855*  3872*  3898*  3915*  3935*  3959*  3963*  3973*  3979*  4038*  4058*  4117*  4141*
                            4145*  4155*  4161*  4172*  4239*  4259*  4318*  4342*  4346*  4356*  4362*  4373*  4438*
                            4458*  4501*  4505*  4515*  4520*  4537*  4554*  4616*  4628*  4645*  4669*  4673*  4683*
                            4688*  4738*  4750*  4767*  4791*  4795*  4805*  4810*  4869*  4883*  4893*  4896*  4903*
                            4909*  4926*  4948*  4972*  4976*  4986*  4991*  5049*  5063*  5073*  5076*  5083*  5089*
                            5106*  5128*  5152*  5156*  5166*  5171*  5226*  5238*  5250*  5278*  5302*  5306*  5316*
                            5321*  5375*  5387*  5399*  5427*  5451*  5455*  5465*  5470*  5528*  5540*  5556*  5588*
                            5598*  5601*  5608*  5617*  5641*  5645*  5655*  5660*  5717*  5729*  5745*  5777*  5787*
                            5790*  5797*  5806*  5830*  5834*  5844*  5849*  5906*  5914*  5921*  5953*  5963*  5966*
                            5973*  5982*  6006*  6010*  6020*  6025*  6037*  6059*  6069*  6072*  6079*  6088*  6112*
                            6116*  6126*  6131*  6207*  6211*  6221*  6226*  6290*  6302*  6380*  6384*  6394*  6400*
                            6411*  6492*  6504*  6582*  6586*  6596*  6602*  6613*  6715*  6724*  6736*  6758*  6782*
                            6786*  6796*  6801*  6884*  6893*  6905*  6927*  6951*  6955*  6965*  6970*  7049*  7080*
                            7084*  7094*  7099*  7186*  7217*  7221*  7231*  7236*  7294*  7327*  7353*  7377*  7381*
                            7391*  7396*  7442*  7475*  7501*  7525*  7529*  7539*  7544*  7610*  7613*  7619*  7624*
                            7630*  7640*  7643*  7650*  7659*  7683*  7687*  7697*  7702*  7824*  7827*  7833*  7838*
                            7844*  7854*  7857*  7864*  7873*  7897*  7901*  7911*  7916*  8029*  8043*  8067*  8071*
                            8081*  8087*  8143*  8157*  8181*  8185*  8195*  8201*  8270*  8277*  8311*  8331*  8355*
                            8359*  8369*  8374*  8435*  8442*  8476*  8496*  8520*  8524*  8534*  8539*  10065  10069
                           10074  10077  10080  10088
PWRVEC= 000024            1191#
RAW     = 000020          1381#
RDCHR   = 104410          9215   9407#
RDLIN   = 104411          9288   9408#
RDOCT   = 104412          1992   2048   2074   9409#
RDY     = 000200          1207#
RELERR  001250            1567#  2412*  2440*  2649*  2677*  2902*  2930*  3081*  3109*  3176*  3204*  3219   3321*
                            3349*  3364   3453*  3481*  3533*  3561*  3736*  3764*  3940*  3968*  4122*  4150*  4323*
                            4351*  4482*  4510*  4525   4650*  4678*  4772*  4800*  4888*  4953*  4981*  5068*  5133*
                            5161*  5283*  5311*  5432*  5460*  5593*  5622*  5650*  5782*  5811*  5839*  5958*  5987*
                            6015*  6064*  6093*  6121*  6188*  6216*  6361*  6389*  6563*  6591*  6763*  6791*  6932*
                            6960*  7061*  7089*  7198*  7226*  7358*  7386*  7506*  7534*  7635*  7664*  7692*  7849*
                            7878*  7906*  8048*  8076*  8162*  8190*  8336*  8364*  8501*  8529*
RELOK   = 000001          1931#  2441   2678   2931   3110   3205   3350   3482   3562   3578#  3765   3969   4151
                            4352   4511   4679   4801   4982   5162   5312   5461   5651   5840   6016   6122   6217
                            6390   6592   6792   6961   7090   7227   7387   7535   7693   7907   8077   8191   8365
                            8530
RESREG= 104414            9411#
RESVEC= 000010            1186#
```

RMR = 000004 1283#
RPAS = 000016 1454# 5904* 6180* 7617* 7831* 8022 8032 8136 8146
RPBA = 000004 1449#
RPCA = 000034 1461# 2373 2497 2610 2734
RPCC = 000036 1462# 2372 2502 2609 2739
RPCS1 = 000000 1447# 2808* 2843 2844 2861* 2987* 3022 3023 3040* 3172* 3317* 3445* 3525*
3645* 3732* 3765* 3766* 3849* 3936* 3969* 3970* 4118* 4151* 4152* 4319* 4352*
4353* 4511* 4512* 4601* 4602* 4605* 4606* 4614 4646* 4679* 4680* 4723* 4724*
4727* 4728* 4736 4768* 4801* 4802* 4853* 4854* 4857* 4858* 4884* 4949* 4982*
4983* 5033* 5034* 5037* 5038* 5064* 5129* 5162* 5163* 5211* 5212* 5215* 5216*
5224 5256 5257 5279* 5312* 5313* 5360* 5361* 5364* 5365* 5373 5405 5406
5428* 5461* 5462* 5513* 5514* 5517* 5518* 5557* 5571 5572 5577* 5589* 5618*
5651* 5652* 5702* 5703* 5706* 5707* 5746* 5760 5761 5766* 5778* 5807* 5840*
5841* 5888* 5889* 5892* 5893* 5954* 5983* 6016* 6017* 6060* 6089* 6122* 6123*
6167* 6168* 6171* 6172* 6217* 6218* 6264* 6265* 6268* 6269* 6390* 6391* 6432*
6433* 6466* 6467* 6470* 6471* 6592* 6593* 6634* 6635* 6675* 6676* 6679* 6680*
6694* 6696 6697 6716* 6751* 6759* 6792* 6793* 6844* 6845* 6848* 6849* 6863*
6865 6866 6885* 6920* 6928* 6961* 6962* 7010* 7011* 7014* 7015* 7090* 7091*
7147* 7148* 7151* 7152* 7227* 7228* 7287* 7288* 7291* 7292* 7300* 7354* 7387*
7388* 7435* 7436* 7439* 7440* 7448* 7502* 7535* 7536* 7593* 7594* 7597* 7598*
7631* 7660* 7693* 7694* 7722 7723 7728* 7739 7740 7745* 7756 7757 7762*
7807* 7808* 7811* 7812* 7845* 7874* 7907* 7908* 7936 7937 7942* 7953 7954
7959* 7970 7971 7976* 8010* 8011* 8014* 8015* 8044* 8077* 8078* 8124* 8125*
8128* 8129* 8158* 8191* 8192* 8254* 8255* 8258* 8259* 8282* 8332* 8365* 8366*
8419* 8420* 8423* 8424* 8447* 8497* 8530* 8531*
RPCS2 = 000010 1451# 2126* 2130 2131 2146* 2147* 2151 2152 2167* 2172* 2189* 2210* 2242*
2278* 2280* 2335* 2338* 2350* 2366* 2416* 2420* 2432* 2437* 2572* 2575* 2587*
2603* 2653* 2657* 2669* 2674* 2795* 2813* 2825* 2882* 2906* 2910* 2922* 2927*
2974* 2992* 3004* 3061* 3085* 3089* 3101* 3106* 3161* 3170* 3180* 3184* 3196*
3201* 3222* 3239* 3263* 3306* 3315* 3325* 3329* 3341* 3346* 3367* 3384* 3408*
3438* 3457* 3461* 3473* 3478* 3518* 3537* 3541* 3553* 3558* 3614* 3617* 3629*
3650* 3667* 3688* 3693* 3710* 3730* 3740* 3744* 3756* 3761* 3818* 3821* 3833*
3854* 3871* 3892* 3897* 3914* 3934* 3944* 3948* 3960* 3965* 4022* 4025* 4030*
4037* 4057* 4082* 4090* 4091* 4116* 4126* 4130* 4142* 4147* 4171* 4223* 4226*
4231* 4238* 4258* 4283* 4291* 4292* 4317* 4327* 4331* 4343* 4348* 4372* 4422*
4425* 4430* 4437* 4457* 4486* 4490* 4502* 4507* 4531* 4536* 4553* 4599* 4603*
4612* 4615* 4627* 4644* 4654* 4658* 4670* 4675* 4721* 4725* 4734* 4737* 4749*
4766* 4776* 4780* 4792* 4797* 4851* 4855* 4864* 4868* 4882* 4892* 4895* 4908*
4925* 4947* 4957* 4961* 4973* 4978* 5031* 5035* 5044* 5048* 5062* 5072* 5075*
5088* 5105* 5127* 5137* 5141* 5153* 5158* 5209* 5213* 5222* 5225* 5237* 5249*
5277* 5287* 5291* 5303* 5308* 5358* 5362* 5371* 5374* 5386* 5398* 5426* 5436*
5440* 5452* 5457* 5511* 5515* 5524* 5527* 5539* 5555* 5587* 5597* 5600* 5616*
5626* 5630* 5642* 5647* 5700* 5704* 5713* 5716* 5728* 5744* 5776* 5786* 5789*
5805* 5815* 5819* 5831* 5836* 5886* 5890* 5898* 5905* 5913* 5920* 5952* 5962*
5965* 5981* 5991* 5995* 6007* 6012* 6036* 6058* 6068* 6071* 6087* 6097* 6101*
6113* 6118* 6165* 6169* 6173* 6192* 6196* 6208* 6213* 6262* 6266* 6274* 6289*
6301* 6365* 6369* 6381* 6386* 6410* 6430* 6464* 6468* 6476* 6491* 6503* 6567*
6571* 6583* 6588* 6612* 6632* 6673* 6677* 6685* 6713* 6714* 6723* 6735* 6757*
6767* 6771* 6783* 6788* 6842* 6846* 6854* 6882* 6883* 6892* 6904* 6926* 6936*
6940* 6952* 6957* 7008* 7012* 7021* 7048* 7065* 7069* 7081* 7086* 7145* 7149*
7158* 7185* 7202* 7206* 7218* 7223* 7285* 7289* 7293* 7326* 7352* 7362* 7366*
7378* 7383* 7433* 7437* 7441* 7474* 7500* 7510* 7514* 7526* 7531* 7591* 7595*
7602* 7609* 7612* 7618* 7623* 7629* 7639* 7642* 7658* 7668* 7672* 7684* 7689*
7805* 7809* 7816* 7823* 7826* 7832* 7837* 7843* 7853* 7856* 7872* 7882* 7886*
7898* 7903* 8008* 8012* 8016* 8019* 8028* 8042* 8052* 8056* 8068* 8073* 8122*
8126* 8130* 8133* 8142* 8156* 8166* 8170* 8182* 8187* 8252* 8256* 8265* 8269*

```
                8276* 8330* 8340* 8344* 8356* 8361* 8417* 8421* 8430* 8434* 8441* 8495* 8505*
                8509* 8521* 8526*
RPDA  = 000006  1450# 2378  2472  2615  2709
RPDB  = 000022  1456#
RPDS1 = 000012  1452# 2128  2149  2213  2214  2228  2229  2245  2246  2260  2261  2337* 2341
                2343  2352  2385  2413  2417  2421  2574* 2578  2580  2599  2622  2650  2654
                2658  2816  2817  2828  2829  2868  2869  2888  2903  2907  2911  2995  2996
                3007  3008  3047  3048  3067  3082  3086  3090  3163* 3177  3181  3185  3225
                3226  3242  3243  3265  3308* 3322  3326  3330  3370  3371  3387  3388  3410
                3454  3458  3462  3534  3538  3542  3616* 3620  3622  3631  3653  3654  3670
                3671  3696  3697  3713  3714  3737  3741  3745  3820* 3824  3826  3835  3857
                3858  3874  3875  3900  3901  3917  3918  3941  3945  3949  4026  4031  4040
                4041  4060  4061  4084* 4097  4098  4123  4127  4131  4174  4175  4227  4232
                4241  4242  4261  4262  4285* 4298  4299  4324  4328  4332  4375  4376  4426
                4431  4440  4441  4460  4461  4483  4487  4491  4539  4540  4556  4557  4600*
                4604* 4618  4620  4629  4651  4655  4659  4722* 4726* 4740  4742  4751  4773
                4777  4781  4852* 4856* 4866* 4874* 4890  4894  4897  4911  4912  4928  4929
                4954  4958  4962  5032* 5036* 5046* 5054* 5070  5074  5077  5091  5092  5108
                5109  5134  5138  5142  5210* 5214* 5228  5230  5239  5284  5288  5292  5359*
                5363* 5377  5379  5388  5433  5437  5441  5512* 5516* 5526* 5530  5532  5541
                5563  5564  5595  5599  5602  5623  5627  5631  5701* 5705* 5715* 5719  5721
                5730  5752  5753  5784  5788  5791  5812  5816  5820  5887* 5891* 5911  5915
                5922  5933  5934  5960  5964  5967  5988  5992  5996  6039  6040  6066  6070
                6073  6094  6098  6102  6166* 6170* 6189  6193  6197  6263* 6267* 6276* 6295
                6308  6309  6341  6342  6362  6366  6370  6413  6414  6431* 6465* 6469* 6478*
                6497  6510  6511  6543  6544  6564  6568  6572  6615  6616  6633* 6674* 6678*
                6687* 6726  6727  6764  6768  6772  6843* 6847* 6856* 6895  6896  6933  6937
                6941  7009* 7013* 7023* 7047  7050  7062  7066  7070  7146* 7150* 7160* 7184
                7187  7199  7203  7207  7286* 7290* 7305  7312  7313  7333  7334  7359  7363
                7367  7434* 7438* 7453  7460  7461  7481  7482  7507  7511  7515  7592* 7596*
                7604* 7637  7641  7644  7665  7669  7673  7806* 7810* 7818* 7851  7855  7858
                7879  7883  7887  8009* 8013* 8020  8031* 8049  8053  8057  8123* 8127* 8134
                8145* 8163  8167  8171  8253* 8257* 8267* 8275* 8293  8294  8313  8314  8337
                8341  8345  8418* 8422* 8432* 8440* 8458  8459  8478  8479  8502  8506  8510
RPDT  = 000026  1458# 2175  2176  2192  2193  2377  2477  2614  2714
RPEC1 = 000044  1465# 2369  2517  2606  2754
RPEC2 = 000046  1466# 2368  2522  2605  2759
RPER1 = 000014  1453# 2380  2462  2617  2699  4023* 4024* 4028* 4029* 4224* 4225* 4229* 4230*
                4423* 4424* 4428* 4429* 6282* 6327  6328  6484* 6529  6530  6693* 6738  6739
                6862* 6907  6908  7606* 7616* 7620* 7714  7715  7820* 7830* 7834* 7928  7929
                8017* 8018* 8131* 8132*
RPER2 = 000040  1463# 2375  2487  2612  2724  7607* 7615* 7621* 7731  7732  7821* 7829* 7835*
                7945  7946
RPER3 = 000042  1464# 2370  2512  2607  2749  7608* 7614* 7622* 7748  7749  7822* 7828* 7836*
                7962  7963
RPLA  = 000020  1455# 2376  2482  2613  2719
RPMR  = 000024  1457# 2379  2467  2616  2704
RPOF  = 000032  1460# 2374  2492  2611  2729  2862* 3041*
RPSN  = 000030  1459# 2279  2281  2371  2507  2608  2744
RPWC  = 000002  1448# 2081
SAVREG= 104413  9410#
SC    = 100000  1213#
SC1   = 000100  1346#
SC10  = 001000  1349#
SC2   = 000200  1347#
SC20  = 002000  1350#
```

```
SC4    = 000400          1348#
SEIZPT  001236          1562#  2336*  2573*  2803*  2982*  3162*  3307*  3440*  3520*  3615*  3819*  4083*  4284*
                        4613*  4735*  4865*  5045*  5223*  5372*  5525*  5714*  5899*  5947*  6053*  6275*  6477*
                        6686*  6855*  7022*  7159*  7295*  7443*  7603*  7817*  8030*  8144*  8266*  8431*  10074
                       10077  10080  10082  10084  10086  10093  10095
SKI    = 040000          1421#  1431#
STACK  = 001100          1090*  1947   2038   2121   2319   2556   2794   2973   3149   3294   3437   3517   3608
                         3812   4017   4218   4417   4595   4717   4847   5027   5205   5354   5507   5696   5882
                         6161   6258   6460   6669   6838   7004   7141   7281   7429   7587   7801   8004   8118
                         8248   8413
START   002064          1491   1937#
START1  002072          1495   1939#
START2  002100          1938   1940#
STKLMT= 177774          1101#
SWR     001140          1525#  1945   1965*  1967   1973*  1984   2285   8676   8694   8701   8738   8745   8750
                        9067   9106   9153*
SWREG   000176          1476#  1973   1984   9067   9106   9129
SW0    = 000001          1154#
SW00   = 000001          1144#  1154
SW01   = 000002          1143#  1153
SW02   = 000004          1142#  1152
SW03   = 000010          1141#  1151
SW04   = 000020          1140#  1150
SW05   = 000040          1139#  1149
SW06   = 000100          1138#  1148
SW07   = 000200          1137#  1147
SW08   = 000400          1136#  1146
SW09   = 001000          1135#  1145
SW1    = 000002          1153#
SW10   = 002000          1134#
SW11   = 004000          1133#
SW12   = 010000          1132#
SW13   = 020000          1131#
SW14   = 040000          1130#
SW15   = 100000          1129#  2285
SW2    = 000004          1152#
SW3    = 000010          1151#
SW4    = 000020          1150#
SW5    = 000040          1149#
SW6    = 000100          1148#
SW7    = 000200          1147#
SW8    = 000400          1146#
SW9    = 001000          1145#
TAP    = 040000          1335#
TBITVE= 000014           1187#
TDF    = 000040          1364#  1382#
TESTNO  062314           2047   9456#
TEST1   003142           2118   2119   2121#
TEST10  014016           3434   3435   3437#
TEST11  014440           3514   3515   3517#
TEST12  015062           3605   3606   3608#
TEST13  016440           3809   3810   3812#
TEST14  020016           4014   4015   4017#
TEST15  021320           4215   4216   4218#
TEST16  022622           4414   4415   4417#
TEST17  024062           4592   4593   4595#
```

```
TEST2   004542          2316    2317    2319#
TEST20  024750          4714    4715    4717#
TEST21  025636          4844    4845    4847#
TEST22  027016          5024    5025    5027#
TEST23  030176          5202    5203    5205#
TEST24  031206          5351    5352    5354#
TEST25  032216          5504    5505    5507#
TEST26  033424          5693    5694    5696#
TEST27  034632          5879    5880    5882#
TEST3   006142          2553    2554    2556#
TEST30  036626          6158    6159    6161#
TEST31  037322          6255    6256    6258#
TEST32  040562          6457    6458    6460#
TEST33  042022          6666    6667    6669#
TEST34  043112          6835    6836    6838#
TEST35  044202          7001    7002    7004#
TEST36  045016          7138    7139    7141#
TEST37  045632          7278    7279    7281#
TEST4   007542          2791    2792    2794#
TEST40  046614          7426    7427    7429#
TEST41  047576          7584    7585    7587#
TEST42  051204          7798    7799    7801#
TEST43  052612          8001    8002    8004#
TEST44  053474          8115    8116    8118#
TEST45  054362          8245    8246    8248#
TEST46  055330          8410    8411    8413#
TEST5   010700          2970    2971    2973#
TEST6   012036          3146    3147    3149#
TEST7   013026          3291    3292    3294#
TIME    001252          1568#   2328*   2392    2401    2565*   2629    2638    2801*   2980*   3154*   3299*   6287*   6489*
                        7033*   7041*   7055    7170*   7178*   7192    8634*   10095
TIMEA   001256          1570#   2022*   2321*   2392*   2394
TIMEAM  001262          1572#   2323*   2396*   7110
TIMEAP  001260          1571#   2023*   2322*   2395*   7108    8287
TIMEB   001264          1573#   2024*   2558*   2629*   2631
TIMEBM  001270          1575#   2560*   2633*   7247
TIMEBP  001266          1574#   2025*   2559*   2632*   7245    8452
TIMES   001272          1576#   7055*   7108    7110    7192*   7245    7247
TITLE   062010          1979    9419#
TKVEC =  000060         1194#   9052*   9053*
TOLER   056630          2393    2630    8644#
TPVEC =  000064         1195#
TRAPVE=  000034         1193#   1953*   1954*
TRE    = 040000         1212#   5577    5766    7728    7745    7762    7942    7959    7976
TRK1   = 004000         1351#
TRK10  = 040000         1354#
TRK2   = 010000         1352#
TRK20  = 100000         1355#
TRK4   = 020000         1353#
TRTVEC=  000014         1188#
TSTADR  070612          2058    10125#
TSTERR  062362          9463#
TSTNUM  001242          1564#   8734*   10065   10067   10069   10072   10074   10077   10080   10082   10084   10086   10088
                        10090   10091   10093   10095
TST1    003070          2051    2111#   10125
TST1AA  003064          2096#   8596
```

```
TST10   013744          3427#  10132
TST11   014366          3507#  10133
TST12   015010          3598#  10134
TST13   016366          3802#  10135
TST14   017744          4007#  10136
TST15   021246          4208#  10137
TST16   022550          4407#  10138
TST17   024010          4585#  10139
TST2    004470          2309#  10126
TST20   024676          4707#  10140
TST21   025564          4837#  10141
TST22   026744          5017#  10142
TST23   030124          5195#  10143
TST24   031134          5344#  10144
TST25   032144          5497#  10145
TST26   033352          5686#  10146
TST27   034560          5872#  10147
TST3    006070          2546#  10127
TST30   036554          6151#  10148
TST31   037250          6248#  10149
TST32   040510          6450#  10150
TST33   041750          6659#  10151
TST34   043040          6828#  10152
TST35   044130          6994#  10153
TST36   044744          7131#  10154
TST37   045560          7271#  10155
TST4    007470          2784#  10128
TST40   046542          7419#  10156
TST41   047524          7577#  10157
TST41B  050600          7611    7625    7712#
TST42   051132          7708    7791#  10158
TST42B  052206          7825    7839    7926#
TST43   052540          7922    7994#  10159
TST44   053422          8108#  10160
TST45   054210          8238#  10161
TST46   055256          8403#  10162
TST5    010026          2963#  10129
TST6    011764          3139#  10130
TST7    012754          3284#  10131
TUF     = 000100        1365#   1383#
TYPDS   = 104405        8577    8584    8804    9402#
TYPE    = 104401        1979    1991    1996    2004    2010    2016    2028    2036    2047    2054    2069    2072    2073
                        8571    8578    8585    8740    8748    8768    8785    8787    8790    8792    8807    8812    8850
                        8946    9021    9073    9127    9128    9131    9144    9155    9174    9222    9228    9233    9237
                        9242    9243    9245    9248    9252    9317    9319    9398#
TYPOC   = 104402        2071    8776    8800    9130    9399#
TYPON   = 104404        9401#
TYPOS   = 104403        2007    2013    9400#
UNS     = 040000        1295#
UPE     = 020000        1236#
US1     = 000001        1223#
US2     = 000002        1224#
US4     = 000004        1225#
UWR     = 000010        1417#
VUF     = 000002        1416#
VU30    = 010000        1371#
```

```
VV     = 000100  1268# 2419  2423  2444  2450  2656  2660  2681  2687  2871  2909  2913  2934
                 2940  3050  3084  3088  3092  3179  3183  3187  3324  3328  3332  3456  3460
                 3464  3536  3540  3544  3632  3673  3716  3739  3743  3747  3836  3877  3920
                 3943  3947  3951  4125  4129  4133  4326  4330  4334  4489  4493  4630  4653
                 4657  4661  4752  4775  4779  4783  4889  4956  4960  4964  5069  5136  5140
                 5144  5240  5286  5290  5294  5389  5435  5439  5443  5542  5594  5625  5629
                 5633  5731  5783  5814  5818  5822  5936  5959  5990  5994  5998  6042  6065
                 6096  6100  6104  6191  6195  6199  6368  6372  6570  6574  6766  6770  6774
                 6935  6939  6943  7064  7068  7072  7201  7205  7209  7361  7365  7369  7509
                 7513  7517  7636  7667  7671  7675  7850  7881  7885  7889  8051  8055  8059
                 8165  8169  8173  8339  8343  8347  8504  8508  8512
VVSET  = 000001  1931# 2353  2415  2441  2444  2450  2590  2652  2678  2681  2687  2905  2931
                 2934  2940  3042# 3084  3110  3113  3114  3119  3120  3179  3205  3208  3209
                 3214  3215  3324  3350  3353  3354  3359  3360  3456  3482  3485  3486  3491
                 3492  3536  3562  3565  3566  3571  3572  3632  3739  3765  3767  3770  3771
                 3776  3777  3836  3943  3969  3971  3974  3975  3980  3981  4125  4151  4153
                 4156  4157  4162  4163  4326  4352  4354  4357  4358  4363  4364  4511  4513
                 4516  4521  4630  4653  4679  4681  4684  4689  4752  4775  4801  4803  4806
                 4811  4889  4956  4982  4984  4987  4992  5069  5136  5162  5164  5167  5172
                 5240  5286  5312  5314  5317  5322  5389  5435  5461  5463  5466  5471  5542
                 5594  5625  5651  5653  5656  5661  5731  5783  5814  5840  5842  5845  5850
                 5959  5990  6016  6018  6021  6026  6065  6096  6122  6124  6127  6132  6191
                 6217  6222  6227  6390  6392  6395  6396  6401  6402  6592  6594  6597  6598
                 6603  6604  6766  6792  6794  6797  6802  6935  6961  6963  6966  6971  7064
                 7090  7095  7100  7201  7227  7232  7237  7361  7387  7389  7392  7397  7509
                 7535  7537  7540  7545  7636  7667  7693  7695  7698  7703  7850  7881  7907
                 7909  7912  7917  8051  8077  8079  8082  8083  8088  8089  8165  8191  8193
                 8196  8197  8202  8203  8339  8365  8367  8370  8375  8504  8530  8532  8535
                 8540
WAO    = 000002  1427#
WATCH    001254  1569# 2329* 2387  2566* 2624  2802* 2890  2981* 3069  3155* 3267  3300* 3412
                 6288* 6297  6490* 6499  7034* 7035  7042* 7052  7171* 7172  7179* 7189  8287*
                 8305  8452* 8470  8635  8637* 8639*
WCE    = 040000  1237#
WCF    = 000040  1286#
WCU    = 000001  1359# 1377#
WLE    = 004000  1292#
WRL    = 004000  1273#
WRU    = 000400  1367# 1385#
WSU    = 000004  1361# 1379#
$AUTOB   001134  1522# 1988* 9120  9268
$BDADR   001122  1517# 2131* 2132* 2152* 2153* 2176* 2177* 2193* 2194* 2214* 2215* 2229* 2230*
                 2246* 2247* 2261* 2262* 2342* 2343* 2413* 2414* 2461* 2462* 2466* 2467* 2471*
                 2472* 2476* 2477* 2481* 2482* 2486* 2487* 2491* 2492* 2496* 2497* 2501* 2502*
                 2506* 2507* 2511* 2512* 2516* 2517* 2521* 2522* 2579* 2580* 2650* 2651* 2698*
                 2699* 2703* 2704* 2708* 2709* 2713* 2714* 2718* 2719* 2723* 2724* 2728* 2729*
                 2733* 2734* 2738* 2739* 2743* 2744* 2748* 2749* 2753* 2754* 2758* 2759* 2817*
                 2818* 2829* 2830* 2844* 2845* 2869* 2870* 2903* 2904* 2996* 2997* 3008* 3009*
                 3023* 3024* 3048* 3049* 3082* 3083* 3177* 3178* 3226* 3227* 3243* 3244* 3322*
                 3323* 3371* 3372* 3388* 3389* 3454* 3455* 3534* 3535* 3621* 3622* 3654* 3655*
                 3671* 3672* 3697* 3698* 3714* 3715* 3737* 3738* 3825* 3826* 3858* 3859* 3875*
                 3876* 3901* 3902* 3918* 3919* 3941* 3942* 4041* 4042* 4061* 4062* 4098* 4099*
                 4123* 4124* 4175* 4176* 4242* 4243* 4262* 4263* 4299* 4300* 4324* 4325* 4376*
                 4377* 4441* 4442* 4461* 4462* 4483* 4484* 4540* 4541* 4557* 4558* 4619* 4620*
                 4651* 4652* 4741* 4742* 4773* 4774* 4890* 4891* 4912* 4913* 4929* 4930* 4954*
                 4955* 5070* 5071* 5092* 5093* 5109* 5110* 5134* 5135* 5229* 5230* 5257* 5258*
```

```
                    5284*  5285*  5378*  5379*  5406*  5407*  5433*  5434*  5531*  5532*  5564*  5565*  5572*
                    5573*  5595*  5596*  5623*  5624*  5720*  5721*  5753*  5754*  5761*  5762*  5784*  5785*
                    5812*  5813*  5934*  5935*  5960*  5961*  5988*  5989*  6040*  6041*  6066*  6067*  6094*
                    6095*  6189*  6190*  6309*  6310*  6328*  6329*  6342*  6343*  6362*  6363*  6414*  6415*
                    6511*  6512*  6530*  6531*  6544*  6545*  6564*  6565*  6616*  6617*  6697*  6698*  6727*
                    6728*  6739*  6740*  6764*  6765*  6866*  6867*  6896*  6897*  6908*  6909*  6933*  6934*
                    7062*  7063*  7199*  7200*  7313*  7314*  7334*  7335*  7359*  7360*  7461*  7462*  7482*
                    7483*  7507*  7508*  7637*  7638*  7665*  7666*  7715*  7716*  7723*  7724*  7732*  7733*
                    7740*  7741*  7749*  7750*  7757*  7758*  7851*  7852*  7879*  7880*  7929*  7930*  7937*
                    7938*  7946*  7947*  7954*  7955*  7963*  7964*  7971*  7972*  8022*  8023*  8049*  8050*
                    8136*  8137*  8163*  8164*  8294*  8295*  8314*  8315*  8337*  8338*  8459*  8460*  8479*
                    8480*  8502*  8503*  10065  10067  10069  10072  10074  10077  10082
$BDDAT  001126      1519#  2130*  2134   2138   2151*  2155   2159   2175*  2179   2183   2192*  2196   2200
                    2213*  2217   2221   2228*  2232   2236   2245*  2249   2253   2260*  2264   2268   2281*
                    2282   2341*  2345   2352*  2356   2430*  2436*  2442*  2448*  2463*  2468*  2473*  2478*
                    2483*  2488*  2493*  2498*  2503*  2508*  2513*  2518*  2523*  2578*  2582   2589*  2593
                    2667*  2673*  2679*  2685*  2700*  2705*  2710*  2715*  2720*  2725*  2730*  2735*  2740*
                    2745*  2750*  2755*  2760*  2816*  2820   2828*  2832   2836   2843*  2847   2851   2868*
                    2872   2876   2920*  2926*  2932*  2938*  2995*  2999   3007*  3011   3015   3022*  3026
                    3030   3047*  3051   3055   3099*  3105*  3111*  3117*  3194*  3200*  3206*  3212*  3225*
                    3229   3233   3242*  3246   3250   3339*  3345*  3351*  3357*  3370*  3374   3378   3387*
                    3391   3395   3471*  3477*  3483*  3489*  3551*  3557*  3563*  3569*  3620*  3624   3631*
                    3635   3653*  3657   3661   3670*  3674   3678   3696*  3700   3704   3713*  3717   3721
                    3754*  3760*  3768*  3774*  3824*  3828   3835*  3839   3857*  3861   3865   3874*  3878
                    3882   3900*  3904   3908   3917*  3921   3925   3958*  3964*  3972*  3978*  4040*  4044
                    4048   4060*  4064   4068   4097*  4101   4105   4140*  4146*  4154*  4160*  4174*  4178
                    4182   4241*  4245   4249   4261*  4265   4269   4298*  4302   4306   4341*  4347*  4355*
                    4361*  4375*  4379   4383   4440*  4444   4448   4460*  4464   4468   4500*  4506*  4514*
                    4519*  4539*  4543   4547   4556*  4560   4564   4618*  4622   4629*  4633   4668*  4674*
                    4682*  4687*  4740*  4744   4751*  4755   4790*  4796*  4804*  4809*  4897*  4902*  4904
                    4911*  4915   4919   4928*  4932   4936   4971*  4977*  4985*  4990*  5077*  5082*  5084
                    5091*  5095   5099   5108*  5112   5116   5151*  5157*  5165*  5170*  5228*  5232   5239*
                    5243   5256*  5260   5264   5301*  5307*  5315*  5320*  5377*  5381   5388*  5392   5405*
                    5409   5413   5450*  5456*  5464*  5469*  5530*  5534   5541*  5545   5563*  5567   5571*
                    5574   5602*  5607*  5609   5640*  5646*  5654*  5659*  5719*  5723   5730*  5734   5752*
                    5756   5760*  5763   5791*  5796*  5798   5829*  5835*  5843*  5848*  5933*  5937   5941
                    5967*  5972*  5974   6005*  6011*  6019*  6024*  6039*  6043   6047   6073*  6078*  6080
                    6111*  6117*  6125*  6130*  6206*  6212*  6220*  6225*  6308*  6312   6316   6327*  6331
                    6341*  6345   6349   6379*  6385*  6393*  6399*  6413*  6417   6421   6510*  6514   6518
                    6529*  6533   6543*  6547   6551   6581*  6587*  6595*  6601*  6615*  6619   6623   6696*
                    6700   6704   6726*  6730   6738*  6742   6781*  6787*  6795*  6800*  6865*  6869   6873
                    6895*  6899   6907*  6911   6950*  6956*  6964*  6969*  7079*  7085*  7093*  7098*  7216*
                    7222*  7230*  7235*  7312*  7316   7320   7333*  7337   7341   7376*  7382*  7390*  7395*
                    7460*  7464   7468   7481*  7485   7489   7524*  7530*  7538*  7543*  7644*  7649*  7651
                    7682*  7688*  7696*  7701*  7714*  7718   7722*  7725   7731*  7735   7739*  7742   7748*
                    7752   7756*  7759   7858*  7863*  7865   7896*  7902*  7910*  7915*  7928*  7932   7936*
                    7939   7945*  7949   7953*  7956   7962*  7966   7970*  7973   8032*  8033   8066*  8072*
                    8080*  8086*  8146*  8147   8180*  8186*  8194*  8200*  8293*  8297   8313*  8317   8354*
                    8360*  8368*  8373*  8458*  8462   8478*  8482   8519*  8525*  8533*  8538*  10065  10067
                   10069  10072  10074  10077  10082
$BELL   001202      1545#  8740   8760   9073   9262
$CHARC  057644      8852*  8862*  8869   8878*  8883#
$CKSWR  060574      9106#  9406
$CMTAG  001100      1505#  1942   1943   1951   1955   1956   1957
$CM1  = 000001      1537#  1538#
$CM2  = 000002      1537#  1538#
```

```
$CM3  = 000001  1535#  1537
$CM4  = 000005  1538#  1539#  1540#  1541#  1542#  1543#
$CNTLG  061436  9127  9263#
$CNTLU  061431  9144  9237  9262#
$CRLF  001207  1547#  2016  2036  2072  8585  8748  8760  8768  8787  8792  8812  8851  8886
               9155  9242  9262  9322
$DBLK  060312  8987  9021  9029#
$DOAGN  056434  8567  8588  8594#
$DTBL  060302  8990  9025#
$ENDAD  056424  1484  8590#  8755
$ENDCT  056270  1955  8569#
$ENULL  056440  8597#
$EOP  056224  8208  8557#
$EOPCT  056262  1955*  8566#  8570
$ERFLG  001103  1508#  8666  8690  8692  8698*  8719  8735*  8760
$ERMAX  001115  1514#  1958*  2044*  8692  8714*  8719
$ERROR  057122  1951  8732#
$ERRPC  001116  1515#  8742*  8743*  8744  8760  8774  10065  10067  10069  10072  10074  10077  10080
               10082  10084  10086  10088  10090  10091  10093  10095
$ERRTB  001304  1603#  8782
$ERRTY  057254  8747  8767#
$ERTTL  001112  1512#  8582  8586*  8741*  8760
$ESCAP  001200  1544#  1957*  8713*
$FILLC  001156  1533#  8855  8886
$FILLS  001155  1532#  8886
$GDADR  001120  1516#
$GDDAT  001124  1518#  2133*  2136  2140*  2154*  2157  2161*  2178*  2181  2185*  2195*  2198  2202*
               2216*  2219  2223*  2231*  2234  2238*  2248*  2251  2255*  2263*  2266  2270*  2279*
               2282  2344*  2345  2353*  2354  2358  2415*  2445  2451  2581*  2582  2590*  2591
               2595  2652*  2682  2688  2819*  2820  2831*  2834  2838*  2846*  2849  2853*  2871*
               2874  2878*  2905*  2935  2941  2998*  2999  3010*  3013  3017*  3025*  3028  3032*
               3050*  3053  3057*  3084*  3114  3120  3179*  3209  3215  3228*  3231  3235*  3245*
               3248  3252*  3324*  3354  3360  3373*  3376  3380*  3390*  3393  3397*  3456*  3486
               3492  3536*  3566  3572  3623*  3624  3632*  3633  3637  3656*  3659  3663*  3673*
               3676  3680*  3699*  3702  3706*  3716*  3719  3723*  3739*  3771  3777  3827*  3828
               3836*  3837  3841  3860*  3863  3867*  3877*  3880  3884*  3903*  3906  3910*  3920*
               3923  3927*  3943*  3975  3981  4043*  4046  4050*  4063*  4066  4070*  4100*  4103
               4107*  4125*  4157  4163  4177*  4180  4184*  4244*  4247  4251*  4264*  4267  4271*
               4301*  4304  4308*  4326*  4358  4364  4378*  4381  4385*  4443*  4446  4450*  4463*
               4466  4470*  4485*  4516  4521  4542*  4545  4549*  4559*  4562  4566*  4621*  4622
               4630*  4631  4635  4653*  4684  4689  4743*  4744  4752*  4753  4757  4775*  4806
               4811  4889*  4904  4914*  4917  4921*  4931*  4934  4938*  4956*  4987  4992  5069*
               5084  5094*  5097  5101*  5111*  5114  5118*  5136*  5167  5172  5231*  5232  5240*
               5241  5245  5259*  5262  5266*  5286*  5317  5322  5380*  5381  5389*  5390  5394
               5408*  5411  5415*  5435*  5466  5471  5533*  5534  5542*  5543  5547  5566*  5567
               5594*  5609  5625*  5656  5661  5722*  5723  5731*  5732  5736  5755*  5756  5783*
               5798  5814*  5845  5850  5936*  5939  5943*  5959*  5974  5990*  6021  6026  6042*
               6045  6049*  6065*  6080  6096*  6127  6132  6191*  6222  6227  6311*  6314  6318*
               6330*  6331  6344*  6347  6351*  6364*  6396  6402  6416*  6419  6423*  6513*  6516
               6520*  6532*  6533  6546*  6549  6553*  6566*  6598  6604  6618*  6621  6625*  6699*
               6702  6706*  6729*  6730  6741*  6742  6766*  6797  6802  6868*  6871  6875*  6898*
               6899  6910*  6911  6935*  6966  6971  7064*  7095  7100  7201*  7232  7237  7315*
               7318  7322*  7336*  7339  7343*  7361*  7392  7397  7463*  7466  7470*  7484*  7487
               7491*  7509*  7540  7545  7636*  7651  7667*  7698  7703  7717*  7718  7734*  7735
               7751*  7752  7850*  7865  7881*  7912  7917  7931*  7932  7948*  7949  7965*  7966
               8024*  8035  8051*  8083  8089  8138*  8149  8165*  8197  8203  8296*  8297  8316*
```

```
                        8317    8339*   8370    8375    8461*   8462    8481*   8482    8504*   8535    8540    10067   10069
                       10074
$GET42  056414          8587#
$GTSWR  060664          9128#   9404
$HD   = 000000          1074
$HIOCT  061624          9310*   9321#
$ICNT   001104          1509#   2060*   8705*   8706    8708*   8718
$INTAG  001135          1523#   9125*   9156    9268
$ITEMB  001114          1513#   8744*   8760    8771
$LF     001210          1548#   8760    8886    9252    9262    9322
$LKCSB  001214          1553#   8616*
$LKCSR  001212          1552#   8612    8617*
$LKS    001220          1555#   8621    8625*
$LLVEC  001222          1556#   8622
$LPADR  001106          1510#   1959*   2045*   2118*   2316*   2553*   2791*   2970*   3146*   3291*   3434*   3514*   3605*
                        3809*   4014*   4215*   4414*   4592*   4714*   4844*   5024*   5202*   5351*   5504*   5693*   5879*
                        6158*   6255*   6457*   6666*   6835*   7001*   7138*   7278*   7426*   7584*   7798*   8001*   8115*
                        8245*   8410*   8696*   8711*   8716    8718
$LPERR  001110          1511#   1960*   2046*   2119*   2317*   2554*   2792*   2971*   3147*   3292*   3435*   3515*   3606*
                        3810*   4015*   4216*   4415*   4593*   4715*   4845*   5025*   5203*   5352*   5505*   5694*   5880*
                        6159*   6256*   6458*   6667*   6836*   7002*   7139*   7279*   7427*   7585*   7799*   8002*   8027*
                        8116*   8141*   8246*   8411*   8696    8712*   8718
$LPVEC  001216          1554#   8613
$MAIL = ****** U        1977    1984    8711    8750    8839
$MNEW   061454          9131    9266#
$MSWR   061443          9128    9264#
$MXCNT  057120          8709    8718#
$NULL   001154          1531#   8857    8886
$NWTST= 000001          2098#   2100    2291#   2293    2528#   2530    2766#   2768    2947#   2949    3127#   3129    3272#
                        3274    3418#   3420    3498#   3500    3579#   3581    3783#   3785    3988#   3990    4190#   4192
                        4393#   4395    4573#   4575    4695#   4697    4818#   4820    4998#   5000    5179#   5181    5328#
                        5330    5478#   5480    5667#   5669    5857#   5859    6139#   6141    6234#   6236    6436#   6438
                        6639#   6641    6808#   6810    6978#   6980    7115#   7117    7254#   7256    7403#   7405    7552#
                        7554    7766#   7768    7981#   7983    8095#   8097    8217#   8219    8382#   8384
$OCNT   060072          8918#   8947#   8960#
$OMODE  060074          8913#   8917#   8922    8925*   8936*   8962#
$OVER   057104          8677    8697    8707    8715#
$PASS   001100          1506#   2043*   8563*   8564*   8575    8597    8703    8719
$QUES   001206          1546#   8760    8886    9174    9245    9262    9319    9322
$RDCHR  061076          9187#   9407
$RDDEC= ****** U        9410
$RDLIN  061166          9210#   9408
$RDOCT  061466          9283#   9409
$RDSZ = 000007          9203#
$REGAD  001160          1535#
$REGO   001162          1537#
$RESRE  061664          9355#   9411
$RPADR  001300          1586#   2037    2070    2077*   2080    2096    10097
$RPVEC  001302          1587#
$RTNAD  056436          8596#
$R2A  = ****** U        9412
$SAVRE  061626          9339#   9410
$SCOPE  056670          1949    8674#
$SETUP= 000127          1931#   1948    1949    1951    1953    1955    1956    1957    1959    1981    8561    8675    8733
                        8753    8755    9065    9070    9071    9101    9268
$STUP = 177777          1931#
```

```
$SVLAD  057056          8685    8710#
$SVPC = 000200          1482#   1487
$SWR  = 166000          1063#   1074    1079    1080    1081    1082    1083    1084    1085    1543    1544    1545    1956
                        1957    1959    1960    2120    2318    2555    2793    2972    3148    3293    3436    3516    3607
                        3811    4016    4217    4416    4594    4716    4846    5026    5204    5353    5506    5695    5881
                        6160    6257    6459    6668    6837    7003    7140    7280    7428    7596    7800    8003    8117
                        8247    8412    8554    8562    8589    8595    8597    8667    8668    8669    8670    8671    8676
                        8688    8690    8691    8692    8699    8700    8701    8712    8715    8718    8719#   8725    8726
                        8727    8728    8729    8738    8745    8750    8754    8760
$SWRMK= 000000          8671
$TIMES  001176          1543#   1956*   2120*   2318*   2555*   2793*   2972*   3148*   3293*   3436*   3516*   3607*   3811*
                        4016*   4217*   4416*   4594*   4716*   4846*   5026*   5204*   5353*   5506*   5695*   5881*   6160*
                        6257*   6459*   6668*   6837*   7003*   7140*   7280*   7428*   7586*   7800*   8003*   8117*   8247*
                        8412*   8562*   8699*   8706    8709*   8718
$TKB    001146          1528#   9033    9054    9063    9082    9110    9137
$TKCNT  060322          9034#   9049*   9071    9088*   9194    9196*
$TKINT  060332          1980    2041    9049#   9123
$TKQEN= 060331          9038#   9096    9199
$TKQIN  060324          9035#   9050*   9051    9094*   9095*   9096    9098*
$TKQOU  060326          9036#   9051*   9197    9198*   9199    9201*
$TKQSR  060330          9037#   9050    9098    9201
$TKS    001144          1527#   9033    9055#   9078*   9080    9086*   9108    9124*   9134    9158*
$TKSRV  060402          9052    9063#
$TMPO   001164          1538#   2134*   2135*   2136    2155*   2156*   2157    2179*   2180*   2181    2196*   2197*   2198
                        2217*   2218*   2219    2232*   2233*   2234    2249*   2250*   2251    2264*   2265*   2266    2356*
                        2357*   2358    2418*   2419*   2424    2426    2433    2593*   2594*   2595    2655*   2656*   2661
                        2663    2670    2832*   2833*   2834    2847*   2848*   2849    2872*   2873*   2874    2908*   2909*
                        2914    2916    2923    3011*   3012*   3013    3026*   3027*   3028    3051*   3052*   3053    3087*
                        3088*   3093    3095    3102    3182*   3183*   3188    3190    3197    3229*   3230*   3231    3246*
                        3247*   3248    3327*   3328*   3333    3335    3342    3374*   3375*   3376    3391*   3392*   3393
                        3459*   3460*   3465    3467    3474    3539*   3540*   3545    3547    3554    3635*   3636*   3637
                        3657*   3658*   3659    3674*   3675*   3676    3700*   3701*   3702    3717*   3718*   3719    3742*
                        3743*   3748    3750    3757    3839*   3840*   3841    3861*   3862*   3863    3878*   3879*   3880
                        3904*   3305*   3906    3921*   3922*   3923    3946*   3947*   3952    3954    3961    4044*   4045*
                        4046    4064*   4065*   4066    4101*   4102*   4103    4128*   4129*   4134    4136    4143    4178*
                        4179*   4180    4245*   4246*   4247    4265*   4266*   4267    4302*   4303*   4304    4329*   4330*
                        4335    4337    4344    4379*   4380*   4381    4444*   4445*   4446    4464*   4465*   4466    4488*
                        4489*   4494    4496    4503    4543*   4544*   4545    4560*   4561*   4562    4633*   4634*   4635
                        4656*   4657*   4662    4664    4671    4755*   4756*   4757    4778*   4779*   4784    4786    4793
                        4894*   4899    4902    4915*   4916*   4917    4932*   4933*   4934    4959*   4960*   4965    4967
                        4974    5074*   5079    5082    5095*   5096*   5097    5112*   5113*   5114    5139*   5140*   5145
                        5147    5154    5243*   5244*   5245    5260*   5261*   5262    5289*   5290*   5295    5297    5304
                        5392*   5393*   5394    5409*   5410*   5411    5438*   5439*   5444    5446    5453    5545*   5546*
                        5547    5599*   5604    5607    5628*   5629*   5634    5636    5643    5734*   5735*   5736    5788*
                        5793    5796    5817*   5818*   5823    5825    5832    5937*   5938*   5939    5964*   5969    5972
                        5993*   5994*   5999    6001    6008    6043*   6044*   6045    6070*   6075    6078    6099*   6100*
                        6105    6107    6114    6178*   6179*   6180    6194*   6195*   6200    6202    6209    6312*   6313*
                        6314    6345*   6346*   6347    6367*   6368*   6373    6375    6382    6417*   6418*   6419    6514*
                        6515*   6516    6547*   6548*   6549    6569*   6570*   6575    6577    6584    6619*   6620*   6621
                        6700*   6701*   6702    6769*   6770*   6775    6777    6784    6869*   6870*   6871    6938*   6939*
                        6944    6946    6953    7067*   7068*   7073    7075    7082    7204*   7205*   7210    7212    7219
                        7316*   7317*   7318    7337*   7338*   7339    7364*   7365*   7370    7372    7379    7464*   7465*
                        7466    7485*   7486*   7487    7512*   7513*   7518    7520    7527    7641*   7646    7649    7670*
                        7671*   7676    7678    7685    7855*   7860    7863    7884*   7885*   7890    7892    7899    8033*
                        8034*   8035    8054*   8055*   8060    8062    8069    8147*   8148*   8149    8168*   8169*   8174
                        8176    8183    8342*   8343*   8348    8350    8357    8507*   8508*   8513    8515    8522
```

$TMP1 001166
1539# 2354* 2355* 2357 2422* 2423* 2424 2438 2591* 2592* 2594 2659* 2660*
2661 2675 2912* 2913* 2914 2928 3091* 3092* 3093 3107 3186* 3187* 3188
3202 3331* 3332* 3333 3347 3463* 3464* 3465 3479 3543* 3544* 3545 3559
3633* 3634* 3636 3746* 3747* 3748 3762 3837* 3838* 3840 3950* 3951* 3952
3966 4132* 4133* 4134 4148 4333* 4334* 4335 4349 4492* 4493* 4494 4508
4631* 4632* 4634 4660* 4661* 4662 4676 4753* 4754* 4756 4782* 4783* 4784
4798 4963* 4964* 4965 4979 5143* 5144* 5145 5159 5241* 5242* 5244 5293*
5294* 5295 5309 5390* 5391* 5393 5442* 5443* 5444 5458 5543* 5544* 5546
5632* 5633* 5634 5648 5732* 5733* 5735 5821* 5822* 5823 5837 5997* 5998*
5999 6013 6103* 6104* 6105 6119 6198* 6199* 6200 6214 6371* 6372* 6373
6387 6573* 6574* 6575 6589 6773* 6774* 6775 6789 6942* 6943* 6944 6958
7071* 7072* 7073 7087 7208* 7209* 7210 7224 7368* 7369* 7370 7384 7516*
7517* 7518 7532 7674* 7675* 7676 7690 7888* 7889* 7890 7904 8025* 8026*
8034 8058* 8059* 8060 8074 8139* 8140* 8148 8172* 8173* 8174 8188 8346*
8347* 8348 8362 8511* 8512* 8513 8527

$TMP2 001170
1540# 2417* 2418 2430 2442 2444* 2445 2654* 2655 2667 2679 2681* 2682
2907* 2908 2920 2932 2934* 2935 3086* 3087 3099 3111 3113* 3114 3181*
3182 3194 3206 3208* 3209 3326* 3327 3339 3351 3353* 3354 3458* 3459
3471 3483 3485* 3486 3538* 3539 3551 3563 3565* 3566 3741* 3742 3754
3768 3770* 3771 3945* 3946 3958 3972 3974* 3975 4127* 4128 4140 4154
4156* 4157 4328* 4329 4341 4355 4357* 4358 4487* 4488 4500 4514 4516
4655* 4656 4668 4682 4684 4777* 4778 4790 4804 4806 4958* 4959 4971
4985 4987 5138* 5139 5151 5165 5167 5288* 5289 5301 5315 5317 5437*
5438 5450 5464 5466 5627* 5628 5640 5654 5656 5816* 5817 5829 5843
5845 5992* 5993 6005 6019 6021 6098* 6099 6111 6125 6127 6193* 6194
6206 6220 6222 6366* 6367 6379 6393 6395* 6396 6568* 6569 6581 6595
6597* 6598 6768* 6769 6781 6795 6797 6937* 6938 6950 6964 6966 7066*
7067 7079 7093 7095 7203* 7204 7216 7230 7232 7363* 7364 7376 7390
7392 7511* 7512 7524 7538 7540 7669* 7670 7682 7696 7698 7883* 7884
7896 7910 7912 8053* 8054 8066 8080 8082* 8083 8167* 8168 8180 8194
8196* 8197 8341* 8342 8354 8368 8370 8506* 8507 8519 8533 8535 10091

$TMP3 001172
1541# 2421* 2422 2436 2448 2450* 2451 2658* 2659 2673 2685 2687* 2688
2911* 2912 2926 2938 2940* 2941 3090* 3091 3105 3117 3119* 3120 3185*
3186 3200 3212 3214* 3215 3330* 3331 3345 3357 3359* 3360 3462* 3463
3477 3489 3491* 3492 3542* 3543 3557 3569 3571* 3572 3745* 3746 3760
3774 3776* 3777 3949* 3950 3964 3978 3980* 3981 4131* 4132 4146 4160
4162* 4163 4332* 4333 4347 4361 4363* 4364 4491* 4492 4506 4519 4521
4659* 4660 4674 4687 4669 4781* 4782 4796 4809 4811 4962* 4963 4977
4990 4992 5142* 5143 5157 5170 5172 5292* 5293 5307 5320 5322 5441*
5442 5456 5469 5471 5631* 5632 5646 5659 5661 5820* 5821 5835 5848
5850 5996* 5997 6011 6024 6026 6102* 6103 6117 6130 6132 6197* 6198
6212 6225 6227 6370* 6371 6385 6399 6401* 6402 6572* 6573 6587 6601
6603* 6604 6772* 6773 6787 6800 6802 6941* 6942 6956 6969 6971 7070*
7071 7085 7098 7100 7207* 7208 7222 7235 7237 7367* 7368 7382 7395
7397 7515* 7516 7530 7543 7545 7673* 7674 7688 7701 7703 7887* 7888
7902 7915 7917 8057* 8058 8072 8086 8088* 8089 8171* 8172 8186 8200
8202* 8203 8345* 8346 8360 8373 8375 8510* 8511 8525 8538 8540 10091

$TMP4 001174
1542# 2138* 2139* 2140 2159* 2160* 2161 2183* 2184* 2185 2200* 2201* 2202
2221* 2222* 2223 2236* 2237* 2238 2253* 2254* 2255 2268* 2269* 2270 2836*
2837* 2838 2851* 2852* 2853 2876* 2877* 2878 3015* 3016* 3017 3030* 3031*
3032 3055* 3056* 3057 3233* 3234* 3235 3250* 3251* 3252 3378* 3379* 3380
3395* 3396* 3397 3661* 3662* 3663 3678* 3679* 3680 3704* 3705* 3706 3721*
3722* 3723 3865* 3866* 3867 3882* 3883* 3884 3908* 3909* 3910 3925* 3926*
3927 4048* 4049* 4050 4068* 4069* 4070 4105* 4106* 4107 4182* 4183* 4184
4249* 4250* 4251 4269* 4270* 4271 4306* 4307* 4308 4383* 4384* 4385 4448*
4449* 4450 4468* 4469* 4470 4547* 4548* 4549 4564* 4565* 4566 4919* 4920*

```
                    4921    4936*   4937*   4938    5099*   5100*   5101    5116*   5117*   5118    5264*   5265*   5266
                    5413*   5414*   5415    5941*   5942*   5943    6047*   6048*   6049    6316*   6317*   6318    6349*
                    6350*   6351    6421*   6422*   6423    6518*   6519*   6520    6551*   6552*   6553    6623*   6624*
                    6625    6704*   6705*   6706    6873*   6874*   6875    7320*   7321*   7322    7341*   7342*   7343
                    7468*   7469*   7470    7489*   7490*   7491
$TN     = 000047    1063#   1074    2098    2111    2120#   2121    2291    2309    2318#   2319    2528    2546    2555#
                    2556    2766    2784    2793#   2794    2947    2963    2972#   2973    3127    3139    3148#   3149
                    3272    3284    3293#   3294    3418    3427    3436#   3437    3498    3507    3516#   3517    3579
                    3598    3607#   3608    3783    3802    3811#   3812    3988    4007    4016#   4017    4190    4208
                    4217#   4218    4393    4407    4416#   4417    4573    4585    4594#   4595    4695    4707    4716#
                    4717    4818    4837    4846#   4847    4998    5017    5026#   5027    5179    5195    5204#   5205
                    5328    5344    5353#   5354    5478    5497    5506#   5507    5667    5686    5695#   5696    5857
                    5872    5881#   5882    6139    6151    6160#   6161    6234    6248    6257#   6258    6436    6450
                    6459#   6460    6639    6659    6668#   6669    6808    6828    6837#   6838    6978    6994    7003#
                    7004    7115    7131    7140#   7141    7254    7271    7280#   7281    7403    7419    7428#   7429
                    7552    7577    7586#   7587    7588    7708    7766    7791    7800#   7801    7802    7922    7981
                    7994    8003#   8004    8095    8108    8117#   8118    8217    8238    8247#   8248    8382    8403
                    8412#   8413    10175
$TPB    001152      1530#   8875*   8886
$TPFLG  001157      1534#   8833    8886
$TPS    001150      1529#   8873    8886
$TRAP   061722      1953    9375#
$TRAP2  061744      9386#   9397
$TRP    = 000015    9390#   9399#   9400#   9401#   9402#   9403#   9404    9405#   9406    9407#   9408#   9409#   9410#
                    9411#   9412#
$TRPAD  061756      9380    9397#
$TSTNM  001102      1507#   2117*   2315*   2552*   2790*   2969*   3145*   3290*   3433*   3513*   3604*   3808*   4013*
                    4214*   4413*   4591*   4713*   4843*   5023*   5201*   5350*   5503*   5692*   5878*   6157*   6254*
                    6456*   6665*   6834*   7000*   7137*   7277*   7425*   7583*   7797*   8000*   8114*   8244*   8409*
                    8561*   8666    8710*   8715    8719    8734    8737    8760
$TTYIN  061422      9212    9213    9225    9243    9257    9261#
$TYPBN= ****** U    9403
$TYPDS  060076      8975#   9402
$TYPE   057430      8833#   9390    9398
$TYPEC  057600      8854    8861    8868    8873#   8874    9160
$TYPEX  057646      8879    8881    8884#
$TYPOC  057674      8916#   9399
$TYPON  057710      8915    8918#   9401
$TYPOS  057650      8911#   9400
$XTSTR  056702      8679#
$$GET4= 000000      8589#
$OFILL  060073      8912*   8916*   8926    8961#
$40CAT= ****** U    8676    8747
.       = 070740    1470#   1474#   1482    1483#   1485#   1487#   1504#   1549    1946    1959    1960    2045    2046
                    2145    2166    2464    2469    2474    2479    2484    2489    2494    2499    2504    2509    2514
                    2519    2524    2701    2706    2711    2716    2721    2726    2731    2736    2741    2746    2751
                    2756    2761    3220    3365    4055    4075    4256    4276    4455    4475    4526    5580    5769
                    5923    6711    6880    7111    7248    7306    7454    8303    8306    8323    8468    8471    8488
                    8559    8574#   8597    8598#   8718    8719    8760    8815#   8886    9029#   9033    9037#   9038
                    9039#   9261#   9262    9268#   9322
```

```
AA          2320*  2557
BB          2795*  2974
BB1         2795*  2863  3042
CC          3150*  3295
CC1         3438*  3518
CHECK       1063*  2129  2150  2174  2191  2212  2227  2244  2259  2815  2827  2642  2867  2994  3006
            3021   3046  3224  3241  3369  3386  3652  3669  3695  3712  3856  3873  3899  3916  4039
            4059   4096  4173  4240  4260  4297  4374  4439  4459  4538  4555  4910  4927  5090  5107
            5255   5404  5562  5751  5931  6038  6307  6326  6340  6412  6509  6528  6542  6614  6695
            6725   6737  6864  6894  6906  7311  7332  7459  7480  7713  7730  7747  7927  7944  7961
            8292   8312  8457  8477
CLRATA      1063*  4596  4718  4848  5028  5206  5355  5508  5697  5883  6162  6259  6427  6461  6629
            6670   6839  7005  7142  7282  7430  7588  7802  8005  8119  8249  8414
COMMEN      1197*
DD          4596*  4718
EE          4848*  5028
ENDCOM      1197*
ERROR       1091*  2083  2141  2162  2186  2203  2224  2239  2256  2271  2284  2347  2360  2389  2403
            2428   2441  2447  2453  2465  2470  2475  2480  2485  2490  2495  2500  2505  2510  2515
            2520   2525  2584  2597  2626  2640  2665  2678  2684  2690  2702  2707  2712  2717  2722
            2727   2732  2737  2742  2747  2752  2757  2762  2822  2839  2854  2879  2892  2918  2931
            2937   2943  3001  3018  3033  3058  3071  3097  3110  3116  3122  3192  3205  3211  3217
            3236   3253  3269  3337  3350  3356  3362  3381  3398  3414  3469  3482  3488  3494  3549
            3562   3568  3574  3626  3639  3664  3681  3707  3724  3752  3767  3773  3779  3830  3843
            3868   3885  3911  3928  3956  3971  3977  3983  4051  4071  4108  4138  4153  4159  4165
            4185   4252  4272  4309  4339  4354  4360  4366  4386  4451  4471  4498  4513  4518  4523
            4550   4567  4624  4637  4666  4681  4686  4691  4746  4759  4788  4803  4808  4813  4901
            4906   4922  4939  4969  4984  4989  4994  5081  5086  5102  5119  5149  5164  5169  5174
            5234   5247  5267  5299  5314  5319  5324  5383  5396  5416  5448  5463  5468  5473  5536
            5549   5569  5576  5606  5611  5638  5653  5658  5663  5725  5738  5758  5765  5795  5800
            5827   5842  5847  5852  5917  5925  5944  5971  5976  6003  6018  6023  6028  6050  6077
            6082   6109  6124  6129  6134  6204  6219  6224  6229  6299  6319  6333  6352  6377  6392
            6398   6404  6424  6501  6521  6535  6554  6579  6594  6600  6606  6626  6707  6732  6744
            6779   6794  6799  6804  6876  6901  6913  6948  6963  6968  6973  7054  7077  7092  7097
            7102   7112  7191  7214  7229  7234  7239  7249  7323  7344  7374  7389  7394  7399  7471
            7492   7522  7537  7542  7547  7648  7653  7680  7695  7700  7705  7720  7727  7737  7744
            7754   7761  7862  7867  7894  7909  7914  7919  7934  7941  7951  7958  7968  7975  8037
            8064   8079  8085  8091  8151  8178  8193  8199  8205  8299  8319  8352  8367  8372  8377
            8464   8484  8517  8532  8537  8542
ESCAPE      1197*
FF          5206*  5355
FF1         5508*  5697
GETPRI      1197*
GETSWR      1197*  1981
GG          5883*
HEAD        1063*  2111  2309  2546  2784  2963  3139  3284  3427  3507  3598  3802  4007  4208  4407
            4585   4707  4837  5017  5195  5344  5497  5686  5872  6151  6248  6450  6659  6828  6994
            7131   7271  7419  7577  7791  7994  8108  8238  8403
HH          6162*
II          6259*  6461
JJ          6670*  6839
KK          7005*  7142
LL          7588*  7802
LL1         7588*  7708  7922
MM          7282*  7430
MMD         4418*
```

DZRJEA.CMB 02-NOV-76 18:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
MM1     4018#  4219
MORETA  1497#  1550
MSG     2095#  2100  2290#  2293  2527#  2530  2765#  2768  2946#  2949  3126#  3129  3271#  3274  3417#
        3420   3497#  3500   3578#  3581   3782#  3785   3987#  3990   4189#  4192   4392#  4395   4572#  4575
        4694#  4697   4817#  4820   4997#  5000   5178#  5181   5327#  5330   5477#  5480   5666#  5669   5856#
        5859   6138#  6141   6233#  6236   6435#  6438   6638#  6641   6807#  6810   6977#  6980   7114#  7117
        7253#  7256   7402#  7405   7551#  7554   7765#  7768   7980#  7983   8094#  8097   8216#  8219   8381#
        8384
MULT    1197#
NEUTRA  1063#  2409   2646   2898   3077   3173   3318   3450   3530   3733   3937   4119   4320   4479   4647
        4769   4950   5130   5280   5429   5619   5808   5984   6090   6185   6358   6560   6760   6929   7058
        7195   7355   7503   7661   7875   8045   8159   8333   8498
NEWTST  1197#  2098   2291   2528   2766   2947   3127   3272   3418   3498   3579   3783   3988   4190   4393
        4573   4695   4818   4998   5179   5328   5478   5667   5857   6139   6234   6436   6639   6808   6978
        7115   7254   7403   7552   7766   7981   8095   8217   8382
NN      8249#  8414
OO      3609#  3813
POP     1197#  9016   9311   9360
PUSH    1197#  8975   9285   9340
RELEAS  1063#  3167   3312   3727   3931   4113   4314   4641   4763   4879   4944   5059   5124   5274   5423
        5584   5613   5773   5802   5949   5978   6055   6084   6754   6923   7349   7497   7626   7655   7840
        7869   8038   8152   8327   8492
REPORT  1197#
RR      8005#  8119
SCOPE   1092#  2288   2526   2763   2945   3124   3270   3415   3496   3576   3781   3985   4188   4389   4570
        4693   4815   4996   5176   5326   5475   5665   5854   6136   6231   6434   6636   6806   6975   7113
        7250   7401   7549   7707   7921   8093   8207   8380   8545
SEIZE   1063#  2332   2569   3158   3303   3611   3815   4079   4280   4609   4731   4861   5041   5219   5368
        5521   5710   6271   6473   6682   6851   7018   7155   7599   7813   8262   8427
SELECT  1063#  2126   2147   2172   2189   2210   2242   2338   2349   2366   2575   2586   2603   2795   2813
        2825   2882   2974   2992   3004   3061   3170   3222   3239   3262   3315   3367   3384   3407   3438
        3518   3617   3628   3650   3667   3693   3710   3730   3821   3832   3854   3871   3897   3914   3934
        4037   4057   4116   4171   4238   4258   4317   4372   4437   4457   4536   4553   4615   4626   4644
        4737   4748   4766   4868   4882   4892   4895   4908   4925   4947   5048   5062   5072   5075   5088
        5105   5127   5225   5236   5249   5277   5374   5385   5398   5426   5527   5538   5555   5587   5597
        5600   5616   5716   5727   5744   5776   5786   5789   5805   5905   5913   5919   5952   5962   5965
        5981   6035   6058   6068   6071   6087   6289   6300   6410   6491   6502   6612   6714   6722   6735
        6757   6883   6891   6904   6926   7048   7185   7293   7326   7352   7441   7474   7500   7609   7612
        7618   7623   7629   7639   7642   7658   7823   7826   7832   7837   7843   7853   7856   7872   8028
        8042   8142   8156   8269   8276   8330   8434   8441   8495
SETATA  1063#  4019   4220   4419
SETPRI  1197#  9190
SETTRA  9390#  9399   9400   9401   9402   9404   9406   9407   9408   9409   9410   9411
SETUP   1197#  1941
SKIP    1197#
SLASH   1197#
SPACE   1197#
STARS   1197#  1198   1202   1244   1248   1441   1445   1480   1500   1549   1580   1584   1931   1935   2089
        2093   2098   2110   2123   2169   2207   2275   2291   2308   2325   2331   2363   2382   2398   2405
        2456   2528   2545   2562   2568   2600   2619   2635   2642   2693   2766   2783   2798   2805   2810
        2858   2864   2885   2895   2947   2962   2977   2984   2989   3037   3043   3064   3074   3127   3138
        3151   3157   3166   3258   3272   3283   3296   3302   3311   3403   3418   3426   3442   3447   3498
        3506   3522   3527   3579   3597   3610   3642   3647   3685   3690   3783   3801   3814   3846   3851
        3889   3894   3988   4006   4018   4034   4078   4087   4093   4112   4168   4190   4207   4219   4235
        4279   4288   4294   4313   4369   4393   4406   4418   4434   4478   4528   4533   4573   4584   4608
        4640   4695   4706   4730   4762   4818   4836   4860   4871   4876   4943   4998   5016   5040   5051
```

DZRJEA.CMB 02-NOV-76 18:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
            5056    5123    5179    5194    5218    5252    5271    5328    5343    5367    5401    5420    5478    5496    5520
            5552    5559    5583    5667    5685    5709    5741    5748    5772    5857    5871    5895    5901    5908    5928
            6032    6139    6150    6175    6182    6234    6247    6270    6279    6284    6292    6304    6323    6337    6357
            6407    6436    6449    6472    6481    6486    6494    6506    6525    6539    6559    6609    6639    6658    6681
            6690    6719    6748    6753    6808    6827    6850    6859    6888    6917    6922    6978    6993    7017    7026
            7030    7038    7044    7057    7105    7115    7130    7154    7163    7167    7175    7191    7194    7242    7254
            7270    7297    7302    7308    7329    7348    7403    7418    7445    7450    7456    7477    7496    7552    7576
            7766    7790    7981    7993    8095    8107    8210    8214    8217    8237    8261    8272    8279    8284    8289
            8308    8326    8382    8402    8426    8437    8444    8449    8454    8473    8491    8549    8600    8604    8655
            8659    8663    8721    8762    8818    8888    8965    9032    9101    9116    9179    9203    9271    9324    9369
            9413    9417    9479    9483   10117   10121
SWITCH      1063#   4885    5065    5590    5779    5955    6061    7632    7846
SWRSU       1197#   1961#
TESTAG      1063#   2121    2319    2556    2794    2973    3149    3294    3437    3517    3608    3812    4017    4218    4417
            4595    4717    4847    5027    5205    5354    5507    5696    5882    6161    6258    6460    6669    6838    7004
            7141    7281    7429    7587    7801    8004    8118    8248    8413
TIMER       1063#   2324    2561    2797    2976    3150    3295    6283    6485    7029    7037    7166    7174
TRMTRP      9390#
TYPBIN      1197#
TYPDEC      1197#   8575    8582    8802
TYPNAM      1197#
TYPNUM      1197#
TYPOCS      1197#   2005    2011
TYPOCT      1197#   8774    8799    9129
TYPTXT      1197#   8571    8578
USEREO      8547#   8558
$$CMRE      1498#   1537
$$CMTM      1498#   1538    1539    1540    1541    1542
$$ESCA      1197#
$$NEWT      1197#   2098    2291    2528    2766    2947    3127    3272    3418    3498    3579    3783    3988    4190    4393
            4573    4695    4818    4998    5179    5328    5478    5667    5857    6139    6234    6436    6639    6808    6978
            7115    7254    7403    7552    7766    7981    8095    8217    8382
$$SET       9390#   9399    9400    9401    9402    9404    9406    9407    9408    9409    9410    9411
$$SKIP      1197#
.EQUAT      1063#   1087
.HEADE      1063#   1064
.SETUP      1063#   1931
.SWRHI      1063#   1075
.SWRLO      1063#   1085#
.$ACT1      1063#   1478
.$CATC      1063#   1468
.$CMTA      1063#   1498
.$EOP       1063#   8547
.$ERRO      1063#   8719
.$ERRT      1063#   8760
.$RDOC      1063#   9269
.$READ      1063#   9030
.$SAVE      1063#   9322
.$SCOP      1063#   8661
.$TRAP      1063#   9367
.$TYPD      1063#   8963
.$TYPE      1063#   8816
.$TYPO      1063#   8886

. ABS.  070740     000
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

 NOW.SEQ/SOL/CRF/LI:ME/NL:TOC:MC:MD:CND=DZRJEA.CMB
RUN-TIME:  121 139 15 SECONDS
RUN-TIME RATIO: 1656/276=5.9
CORE USED:  32K  (63 PAGES)
```

C01

Spooler runtime 34 Seconds, 143 KCS, 1092 disk reads, 4 disk writes, 208 pages