RP04/6

DUAL CONTROLLER LOGIC

CZRJEB0

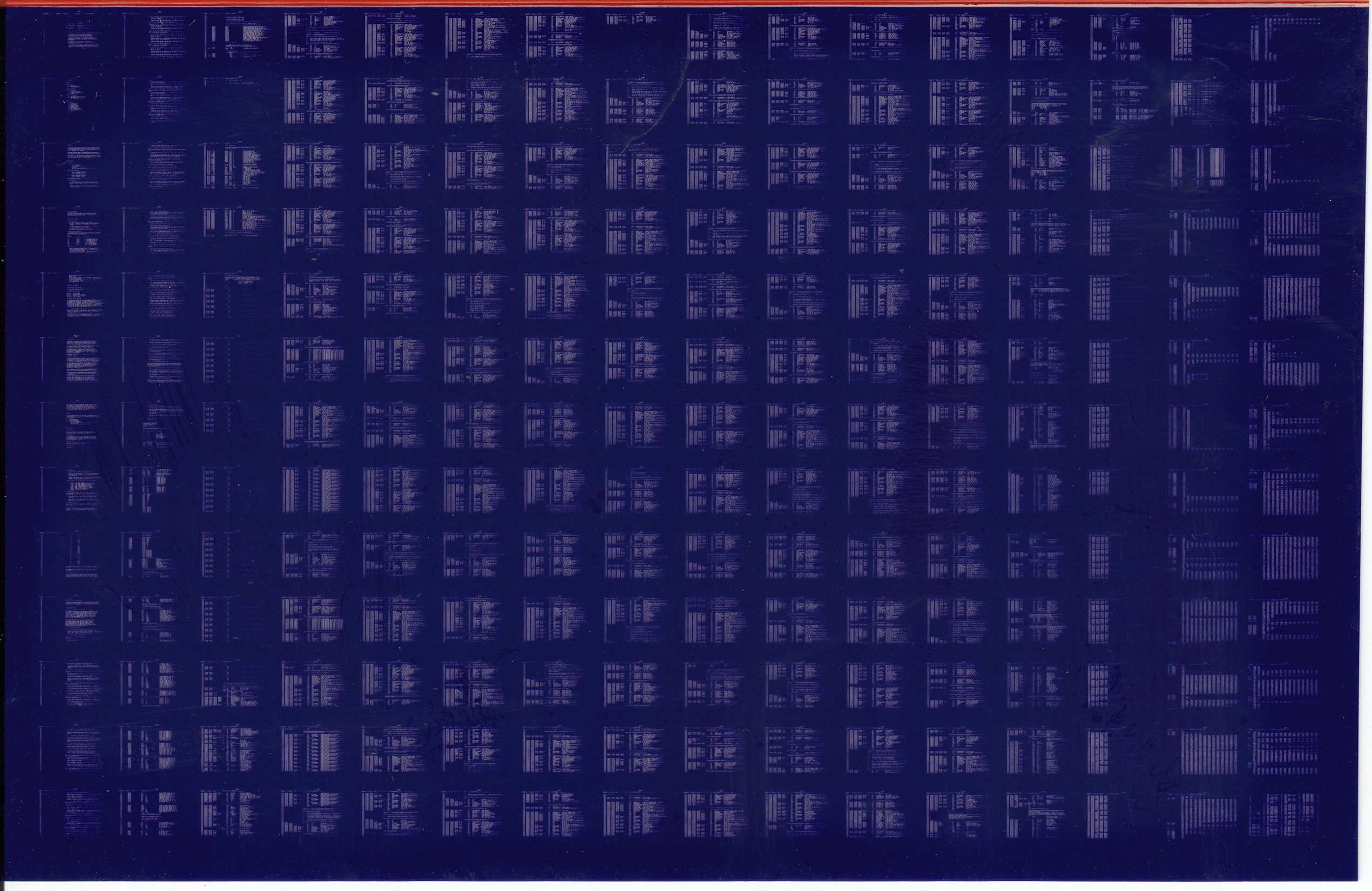
AH-9201B-MC

FICHE 1 OF 2

JAN 1978

digital

MADE IN USA



RP04/6

DUAL CONTROLLER LOGIC

CZRJEB0

AH-9201B-MC

FICHE 2 OF 2

JAN 1978

digital

MADE IN USA

.REM @

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

PRODUCT CODE: AC-92008-MC

PRODUCT NAME: CZRJEB0 RP04/5/6 DUAL CONTROLLER LOGIC TEST

DATE CREATED: DECEMBER 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: C. HESS

CONTENTS

1. ABSTRACT

2. REQUIREMENTS

    2.1 EQUIPMENT
    2.2 PRELIMINARY PROGRAMS
    2.3 OTHER PROGRAMS

3. LOADING PROCEDURES

4. STARTING PROCEDURES

    4.1 STARTING ADDRESSES
    4.2 UNIBUS & VECTOR ADDRESSES
    4.3 OPERATOR ACTION

5. OPERATING PROCEDURES

    5.1 OPERATIONAL SWITCH SETTINGS
    5.2 'SOFTWARE' SWITCH REGISTER
    5.3 TEST SELECTION
    5.4 DUAL PORT TEST CABLE CONNECTION

6. ERRORS

7. MISCELLANEOUS

    7.1 RESTRICTIONS
    7.2 LIMITATIONS
    7.3 EXECUTION TIME
    7.4 STACK POINTER
    7.5 SUBROUTINE CALLS
    7.6 REQUIRED TESTS
    7.7 DISK SURFACE USAGE
    7.8 TEST ITERATIONS
    7.9 LOOP ON ERROR OPTION
    7.10 SPECIAL M7775 'DP' BOARD TESTS

8. TEST DESCRIPTIONS

9. PROGRAM LISTING

1. ABSTRACT

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

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

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

2 REQUIREMENTS

2.1 EQUIPMENT

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

2.2 PRELIMINARY PROGRAMS

RP04/5/6 DISKLESS CONTROLLER TEST
PART 1 (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 200 (8). THE PROGRAM WILL USE THE CURRENT DRIVE (DCL) ADDRESS.

C. THE PROGRAM CAN BE STARTED AT LOCATION 204 (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), 204(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 200.

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 22 - TEST PORT REQUEST FROM PORT 'B'

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

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

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

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

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

8 21 TEST 23 - TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'

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

REQUEST'.

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

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

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

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

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 THRCUGH 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'. VERIF, 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  CZRJEB0, DL CTRLR LGC
                        ;*COPYRIGHT (C) 1976,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY C. HESS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*

                        .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
```

```
                        ;;CODE FOR CARRIAGE RETURN
                        ;;CODE FOR CARRIAGE RETURN-LINE FEED
                        ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
                        ;;STACK LIMIT REGISTER
                        ;;PROGRAM INTERRUPT REQUEST REGISTER
                        ;;HARDWARE SWITCH REGISTER
                        ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;GENERAL REGISTER
                        ;;STACK POINTER
                        ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
                        ;;PRIORITY LEVEL 0
                        ;;PRIORITY LEVEL 1
                        ;;PRIORITY LEVEL 2
                        ;;PRIORITY LEVEL 3
                        ;;PRIORITY LEVEL 4
                        ;;PRIORITY LEVEL 5
                        ;;PRIORITY LEVEL 6
                        ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
```

```
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
000002                          BIT01=  2
000001                          BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

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

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

                                .SBTTL  RH11 REGISTERS

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

                                ;CONTROL AND STATUS REGISTER 1 (RPCS1)

000100                          IE=     100                     ;INTERRUPT ENABLE (BIT #6)
000200                          RDY=    200                     ;READY (BIT #7)
000400                          A16=    400                     ;HIGH ORDER BUS ADDRESS BIT (BIT #8)
```

```
                        001000          A17=    1000            ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
                        002000          PSEL=   2000            ;PORT SELECT (BIT #10)
                        020000          MCPE=   20000           ;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)

                                        ;DFS=   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)
```

```
                        ;GO REVERSE (BIT #3)
                        ;DIFFERENCE LESS THAN 64 (BIT #4)
                        ;DIFFERENCE EQUALS 1 (BIT #5)
                        ;VOLUME VALID (BIT #6)
                        ;DRIVE READY (BIT #7)
                        ;DRIVE PRESENT (BIT #8)
                        ;PROGRAMABLE (BIT #9)
                        ;LAST SECTOR TRANSFERRED (BIT #10)
                        ;WRITE LOCK (BIT #11)
                        ;MEDIUM ON-LINE (BIT #12)
                        ;POSITIONING OPERATION IN PROGRESS (BIT #13)
                        ;COMPOSITE ERROR (BIT #14)
                        ;ATTENTION ACTIVE (BIT #15)

                        ;ERROR REGISTER #01 (RPER1) (#02)

                        ;ILLEGAL FUNCTION (BIT #0)
                        ;ILLEGAL REGISTER (BIT #1)
                        ;REGISTER MODIFICATION REFUSED (BIT #2)
                        ;PARITY ERROR (BIT #3)
                        ;FORMAT ERROR (BIT #4)
                        ;WRITE CLOCK FAIL (BIT #5)
                        ;ECC HARD ERROR (BIT #6)
                        ;HEADER COMPARE ERROR (BIT #7)
                        ;HEADER CRC ERROR (BIT #8)
                        ;ADDRESS OVERFLOW ERROR (BIT #9)
                        ;INVALID ADDRESS ERROR (BIT #10)
                        ;WRITE LOCK ERROR (BIT #11)
                        ;DRIVE TIMING ERROR (BIT #12)
                        ;OPERATION INCOMPLETE (BIT #13)
                        ;DRIVE UNSAFE (BIT #14)
                        ;DATA CHECK ERROR (BIT 15)

                        ;MAINTAINABILITY REGISTER (RPMR)(#03)

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

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

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

                        ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA) (#05)
                        ;(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  003000                      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.

                                        .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
                        $TKS=   177560                  ;;TTY KBD STATUS
                        $TKB=   177562                  ;;TTY KBD BUFFER
                        $TPS=   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                        $TPB=   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001144  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001145  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001146  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001147  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001150  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REGO) WAS OBTAINED
001152  000000          $REGO:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001154  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001156  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001160  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001166  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001170  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001172  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001176  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001177  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001200  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************

                        CR      =       15
                        LF      =       12
001202  172540          $LPCSR: .WORD   172540          ;ADDR OF KW11-P STATUS REGISTER
001204  172542          $LPCSB: .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  062476                          EM1                     ;WRONG DRIVE TYPE
001306  066534                          DH1
001310  070272                          DT1
001312  070542                          DF1

                                ;ERROR 2

001314  062517                          EM2                     ;DRIVE NOT ON LINE
001316  066534                          DH1
001320  070272                          DT1
001322  070542                          DF1

                                ;ERROR 3

001324  062541                          EM3                     ;SERIAL NUMBERS NOT THE SAME
001326  066605                          DH3
001330  070306                          DT3
001332  070542                          DF1

                                ;ERROR 4

001334  062623                          EM4                     ;DRIVE NOT SEIZED BY PORT 'N'
001336  066654                          DH4
001340  070354                          DT7
001342  070555                          DF7

                                ;ERROR 5

001344  062654                          EM5                     ;WRONG STATUS SEEN BY THE SEIZING PORT
001346  066777                          DH5
001350  070322                          DT5
001352  070547                          DF5

                                ;ERROR 6

001354  062722                          EM6                     ;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WA
001356  067247                          DH13
001360  070374                          DT13
```

```
1645  001362  070547                        DF5
1646
1647                                  ;ERROR 7
1648
1649  001364  063022                        EM7                 ;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
1650  001366  067053                        DH7
1651  001370  070354                        DT7
1652  001372  070555                        DF7
1653
1654                                  ;ERROR 10
1655
1656  001374  063103                        EM10                ;REGISTER CONTENTS INCORRECT
1657  001376  066777                        DH5
1658  001400  070322                        DT5
1659  001402  070547                        DF5
1660
1661                                  ;ERROR 11
1662
1663  001404  063133                        EM11                ;CONTROL BUS PARITY ERROR WHILE READING REGISTER
1664  001406  067176                        DH11
1665  001410  070272                        DT1
1666  001412  070542                        DF1
1667
1668                                  ;ERROR 12
1669
1670  001414  063217                        EM12                ;DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
1671  001416  067743                        DH36
1672  001420  070462                        DT37
1673  001422  070570                        DF36
1674
1675                                  ;ERROR 13
1676
1677  001424  063267                        EM13                ;'VOLUME VALID' BIT NOT SET BY READIN PRESET
1678  001426  067247                        DH13
1679  001430  070374                        DT13
1680  001432  070547                        DF5
1681
1682                                  ;ERROR 14
1683
1684  001434  063354                        EM14                ;'VOLUME VALID' SET ON THE OPPOSITE PORT
1685  001436  067247                        DH13
1686  001440  070374                        DT13
1687  001442  070547                        DF5
1688
1689                                  ;ERROR 15
1690
1691  001444  063417                        EM15                ;THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
1692  001446  067053                        DH7
1693  001450  070354                        DT7
1694  001452  070555                        DF7
1695
1696                                  ;ERROR 16
1697
1698  001454  063476                        EM16                ;ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
1699  001456  067053                        DH7
1700  001460  070354                        DT7
```

```
001462  070555                          DF7
                                
                                ;ERROR 17
                                
001464  063551                          EM17                    ;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
001466  067053                          DH7
001470  070354                          DT7
001472  070555                          DF7
                                
                                ;ERROR 20
                                
001474  063630                          EM20                    ;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
001476  067743                          DH36
001500  070462                          DT37
001502  070570                          DF36
                                
                                ;ERROR 21
                                
001504  063710                          EM21                    ;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
001506  067743                          DH36
001510  070462                          DT37
001512  070570                          DF36
                                
                                ;ERROR 22
                                
001514  063763                          EM22                    ;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
001516  067367                          DH22
001520  070412                          DT22
001522  070564                          DF31
                                
                                ;ERROR 23
                                
001524  064050                          EM23                    ;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
001526  067465                          DH23
001530  070424                          DT23
001532  070542                          DF1
```

```
1738                                    ;ERROR 24
1739
1740  001534  064116                            EM24                    ;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
1741  001536  057465                            DH23
1742  001540  070424                            DT23
1743  001542  070542                            DF1
1744
1745
1746                                    ;ERROR 25
1747
1748  001544  064175                            EM25                    ;TIMEOUT ONE-SHOT DID NOT RETRIGGER
1749  001546  067743                            DH36
1750  001550  070452                            DT36
1751  001552  070570                            DF36
1752
1753
1754                                    ;ERROR 26
1755
1756  001554  064240                            EM26                    ;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
1757  001556  067367                            DH22
1758  001560  070412                            DT22
1759  001562  070564                            DF31
1760
1761                                    ;ERROR 27
1762
1763  001564  064325                            EM27                    ;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
1764  001566  067053                            DH7
1765  001570  070354                            DT7
1766  001572  070555                            DF7
1767
1768                                    ;ERROR 30
1769
1770  001574  064403                            EM30                    ;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEJTRAL
1771  001576  067743                            DH36
1772  001600  070452                            DT36
1773  001602  070570                            DF36
1774
1775                                    ;ERROR 31
1776
1777  001604  064500                            EM31                    ;DRIVE NOT SEIZED BY PORT AFTER RELEASE WIITH REQLEST SE
1778  001606  067644                            DH31
1779  001610  070440                            DT31
1780  001612  070564                            DF31
1781
1782                                    ;ERROR 32
1783
1784  001614  064555                            EM32                    ;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
1785  001616  066777                            DH5
1786  001620  070322                            DT5
1787  001622  070547                            DF5
1788
1789                                    ;ERROR 33
1790
1791  001624  064626                            EM33                    ;DRIVE RETURNS TO NEJTRAL IF DRIVE CLEAR GIVEN WHILE DRI
1792  001626  067743                            DH36
1793  001630  070452                            DT36
```

```
1794  001632  070570                        DF36
1795
1796                                  ;ERROR 34
1797
1798  001634  064730                        EM34            ;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
1799  001636  067743                        DH36
1800  001640  070452                        DT36
1801  001642  070570                        DF36
1802
1803                                  ;ERROR 35
1804
1805  001644  065033                        EM35            ;DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
1806  001646  067743                        DH36
1807  001650  070462                        DT37
1808  001652  070570                        DF36
1809
1810                                  ;ERROR 36
1811
1812  001654  065112                        EM36            ;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
1813  001656  067743                        DH36
1814  001660  070452                        DT36
1815  001662  070570                        DF36
1816
1817                                  ;ERROR 37
1818
1819  001664  065164                        EM37            ;DRIVE IS NON-EXISTENT
1820  001666  067743                        DH36
1821  001670  070462                        DT37
1822  001672  070570                        DF36
1823
1824                                  ;ERROR 40
1825
1826  001674  065232                        EM40            ;ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
1827  001676  066534                        DH1
1828  001700  070424                        DT23
1829  001702  070542                        DF1
1830
1831                                  ;ERROR 41
1832
1833  001704  065307                        EM41            ;TIMEOUT CLEARED ATTENTION BIT
1834  001706  067465                        DH23
1835  001710  070424                        DT23
1836  001712  070542                        DF1
1837
1838                                  ;ERROR 42
1839
1840  001714  065351                        EM42            ;DRIVE NOT IN NEUTRAL OR SEIZED
1841  001716  067772                        DH42
1842  001720  070472                        DT42
1843  001722  070573                        DF42
1844
1845                                  ;ERROR 43
1846
1847  001724  065437                        EM43            ;DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
1848  001726  067772                        DH42
1849  001730  070472                        DT42
```

```
1850  001732  070573                        DF42
1851
1852                                  ;ERROR 44
1853
1854  001734  065514                        EM44            ;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
1855  001736  070011                        DH44
1856  001740  070440                        DT31
1857  001742  070564                        DF31
1858
1859                                  ;ERROR 45
1860
1861  001744  065571                        EM45            ;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
1862  001746  066534                        DH1
1863  001750  070272                        DT1
1864  001752  070542                        DF1
1865
1866                                  ;ERROR 46
1867
1868  001754  065650                        EM46            ;CAN'T ACCESS DRIVE THROUGH EITHER PORT
1869  001756  070107                        DH46
1870  001760  070500                        DT46
1871  001762  070564                        DF31
1872
1873                                  ;ERROR 47
1874
1875  001764  065717                        EM47            ;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
1876  001766  067465                        DH23
1877  001770  070424                        DT23
1878  001772  070542                        DF1
1879
1880                                  ;ERROR 50
1881
1882  001774  066005                        EM50            ;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
1883  001776  067247                        DH13
1884  002000  070374                        DT13
1885  002002  070547                        DF5
1886
1887                                  ;ERROR 51
1888
1889  002004  066070                        EM51            ;ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
1890  002006  066777                        DH5
1891  002010  070322                        DT5
1892  002012  070547                        DF5
1893
1894                                  ;ERROR 52
1895
1896  002014  066153                        EM52            ;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
1897  002016  067247                        DH13
1898  002020  070374                        DT13
1899  002022  070547                        DF5
1900
1901                                  ;ERROR 53               ;CAN'T READ ATTN BIT FROM OPPOSITE PORT
1902
1903  002024  066251                        EM53
1904  002026  067465                        DH23
1905  002030  070272                        DT1
```

```
1906  002032  070542                        DF1
1907
1908                                  ;ERROR 54
1909
1910  002034  066332                        EM54                    ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
1911  002036  067367                        DH22
1912  002040  070512                        DT54
1913  002042  070564                        DF31
1914
1915                                  ;ERROR 55
1916
1917  002044  066425                        EM55                    ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
1918  002046  070205                        DH55
1919  002050  070524                        DT55
1920  002052  070575                        DF55
1921
1922                                  ;ERROR 56
1923
1924  002054  066472                        EM56                    ;RH11 DIDN'T RESPOND TO ADDRESSING
1925  002056  070263                        DH56
1926  002060  070536                        DT56
1927  002062  070601                        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     #SWR,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
```

```
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002226  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002232  012737  002266  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002240  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002246  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002254  022777  177777  176656          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002262  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002264  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
002266  012716  002274          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002272  000002                          RTI
002274  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002302  012737  000174  001142          MOV     #DISPREG,DISPLAY
002310  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002314  005227  177777                  INC     #-1             ;FIRST START ?
002320  001002                          BNE     1$              ;BR IF NOT
002322  104401  062010                  TYPE    ,TITLE          ;TYPE PROGRAM NAME
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$:
002362  004737  002754                  JSR     PC,CHANGE       ;CHECK/CHANGE THE RH11 ADDRESS
002366  104401  062107                  TYPE    ,ENTERA         ;ENTER DRIVE ADDRESS
002372  104412                          RDOCT                   ;GET THE ADDRESS
002374  012637  001224                  MOV     (SP)+,PORTA     ;STORE THE ADDRESS
002400  023727  001224  000007          CMP     PORTA,#7        ;SEE IF ADDRESS TOO LARGE
002406  101403                          BLOS    2$              ;BR IF NOT
002410  104401  062137                  TYPE    ,ADRERR         ;TYPE ADDRESS ERROR MESSAGE
002414  000744                          BR      1$              ;TRY AGAIN
002416  013737  001224  001226  2$:     MOV     PORTA,PORTB     ;GENERATE THE PORT B ADDRESS
002424  105237  001226                  INCB    PORTB           ;INCREMENT THE ADDRESS
002430  042737  000016  001226          BIC     #16,PORTB       ;LEAVE BIT 0
002436  013746  001224                  MOV     PORTA,-(SP)     ;PUT PORT A ADDRESS ON THE STACK
002442  042716  177771                  BIC     #^C6,(SP)       ;SAVE BITS 1 & 2
002446  052637  001226                  BIS     (SP)+,PORTB     ;SET BITS 1 & 2 IN PORT B ADDRESS
002452  104401  062161                  TYPE    ,PORTAIS        ;'PORT A ADDRESS IS '
002456  013746  001224                  MOV     PORTA,-(SP)     ;;SAVE PORTA FOR TYPEOUT
                                                                ;;TYPE PORT A ADDRESS
002462  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
002464     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
002465     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
002466  104401  062207                  TYPE    ,PORTBIS        ;'PORT B ADDRESS IS '
002472  013746  001226                  MOV     PORTB,-(SP)     ;;SAVE PORTB FOR TYPEOUT
                                                                ;;TYPE PORT B ADDRESS
002476  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
002500     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
002501     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
002502  104401  001207                  TYPE    ,$CRLF          ;ANOTHER CR-LF
002506  013737  001224  001230          MOV     PORTA,PORTC     ;GENERATE ADDRESS OF DRIVE NOT TESTED
```

```
2018  002514  062737  000006  001230          ADD     #6,PORTC        ;COMPLEMENT SOME BITS
2019  002522  042737  177770  001230          BIC     #^C7,PORTC      ;SAVE ONLY LOWER BITS
2020  002530  013701  001224                  MOV     PORTA,R1        ;USE PORT A ADDRESS AS INDEX
2021  002534  116137  070716  001232          MOVB    ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
2022  002542  005037  001256                  CLR     TIMEA           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2023  002546  005037  001260                  CLR     TIMEAP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2024  002552  005037  001264                  CLR     TIMEB           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2025  002556  005037  001266                  CLR     TIMEBP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
2026  002562  004737  056444                  JSR     PC,CKCLK        ;SETUP CLOCK
2027  002566  000137  002602                  JMP     EXEC            ;CLOCK HAS BEEN STARTED
2028  002572  104401  062235                  TYPE    ,NOCLOCK        ;NO CLOCK ON SYSTEM
2029  002576  000000                  3$:     HALT                    ;FATAL ERROR
2030  002600  000776                          BR      3$              ;INTERLOCK THE HALT
2031
2032                                  ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
2033
2034  002602  000005                  EXEC:   RESET                   ;CLEAR EVERYTHING
2035  002604  005037  177776                  CLR     PS              ;CLEAR THE PROCESSOR STATUS WORD
2036  002610  104401  001207                  TYPE    ,$CRLF          ;CR-LF
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  062304          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  070726          2$:     CMP     R1,MAXTN        ;SEE IF NUMBER GREATER THAN MAXIMUM
2053  002712  003403                          BLE     3$              ;BR IF LESS OR EQUAL
2054  002714  104401  062324                  TYPE    ,BADNO          ;BAD ENTRY
2055  002720  000763                          BR      1$              ;TRY AGAIN
2056  002722  005301                  3$:     DEC     R1              ;DECREMENT ENTRY
2057  002724  006301                          ASL     R1              ;SHIFT IT LEFT
2058  002726  016137  070602  002752          MOV     TSTADR(R1),4$   ;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  062364          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  062444                  TYPE    ,NEWRH11        ;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
2093                                  ;;*****************************************************************
2094
2095
2096  003064  013700  001300          TST1AA: MOV     $RPADR,R0       ;;RESTORE R0 AFTER END OF PASS
2097
2098                                  ;;*****************************************************************
2099                                  ;*TEST 1        DRIVE ACCESS TEST
2100                                  ;*
2101                                  ;*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
2102                                  ;*
2103                                  ;*  A.  SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
2104                                  ;*      DRIVE IS A DUAL PORT RP04, THAT THE DRIVE IS ONLINE (RPDS: HAS
2105                                  ;*      'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL
2106                                  ;*      NUMBER READ THROUGH BOTH PORTS IS THE SAME.
2107                                  ;*
2108                                  ;*  B.  THE TEST IS REPEATED THROUGH BOTH PORTS.
2109                                  ;*
2110                                  ;*****************************************************************
2111  003070                          TST1:
2112  003070  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2113  003074  001406                          BEQ     2$              ;BR IF NOT
2114  003076  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
2115  003100  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
2116  003104  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
2117  003112  112737  000001  001102  2$:     MOVB    #1,$TSTNM       ;TEST NUMBER
2118  003120  012737  003142  001106          MOV     #TEST1,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2119  003126  012737  003142  001110          MOV     #TEST1,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
2120  003134  012737  000001  001176          MOV     #1,$TIMES       ;;DO 1 ITERATION
2121  003142  012706  001100          TEST1:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2122
2123                                  ;;*****************************************************************
2124                                  ;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B
2125
2126  003146  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
2127  003154  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2128  003162  005760  000012                  TST     RPDS1(R0)       ;SEE IF DRIVE (PORT A) PRESENT
2129  003166  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
2130  003172  016037  000010  001126          MOV     RPCS2(R0),$BDDAT  ;GET CONTENTS OF RPCS2
2131  003200  012737  000010  001122          MOV     #RPCS2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2132  003206  060037  001122                  ADD     R0,$BDADR         ;ADD RH11 BASE ADDRESS
2133  003212  005037  001124                  CLR     $GDDAT            ;WHAT REGISTER SHOULD BE
2134  003216  013737  001126  001164          MOV     $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
2135  003224  042737  167777  001164          BIC     #^CNED,$TMP0   ;SAVE SPECIFIED BITS
2136  003232  023737  001124  001164          CMP     $GDDAT,$TMP0      ;COMPARE THE BITS
2137  003240  001414                          BEQ     64$               ;BR IF OK
2138  003242  013737  001126  001174          MOV     $BDDAT,$TMP4      ;COPY 'BAD DATA'
2139  003250  042737  010000  001174          BIC     #NED,$TMP4        ;CLEAR THE MASKED BITS
2140  003256  053737  001174  001124          BIS     $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
2141  003264  104037                          ERROR   37                ;TYPE MESSAGE 37
2142  003266  005137  001244                  COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
2143  003272  000240                  64$:    NOP
2144  003274  005737  001244                  TST     CKERR             ;WAS 'NED' SET ?
2145  003300  001403                          BEQ     .+10              ;BR IF NOT
2146  003302  012760  000040  000010          MOV     #CLR,RPCS2(R0)    ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2147  003310  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B
2148  003316  013737  001226  001234          MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2149  003324  005760  000012                  TST     RPDS1(R0)         ;SEE IF DRIVE (PORT B) PRESENT
2150  003330  005037  001244                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
2151  003334  016037  000010  001126          MOV     RPCS2(R0),$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 TYPEOJT
2162  003426  104037                          ERROR   37                ;TYPE MESSAGE 37
2163  003430  005137  001244                  COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
2164  003434  000240                  66$:    NOP
2165  003436  005737  001244                  TST     CKERR             ;WAS 'NED' SET ?
2166  003442  001403                          BEQ     .+10              ;BR IF NOT
2167  003444  012760  000040  000010          MOV     #CLR,RPCS2(R0)    ;ISSUE MASSBUS INIT TO CLEAR 'NED'
2168
2169                                  ;;*****************************************************************
2170                                  ;CONFIRM THAT DRIVE IS AN RP04/5/6 AND IS DUAL PORT
2171
2172  003452  113760  001224  000010          MOVB    PORTA,RPCS2(R0)   ;SELECT PORT A
2173  003460  013737  001224  001234          MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOJT
2174  003466  005037  001244                  CLR     CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
2175  003472  016037  000026  001126          MOV     RPDT(R0),$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 TYPEOJT
```

```
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$:    NOP
2274
2275
2276                                  ;;*****************************************************************
2277                                  ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
2278  004412  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2279  004420  016037  000030  001124          MOV     RPSN(R0),$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
```

```
                                        ;*      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.
                                        ;*
                                        ;;*****************************************************************
004470                                  TST2:
004470  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
004474  001406                                BEQ     2$              ;BR IF NOT
004476  100002                                BPL     1$              ;BR IF JUST ENTERED TEST
004500  000137  002602                        JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
004504  012737  177777  001274    1$:         MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
004512  112737  000002  001102    2$:         MOVB    #2,$TSTNM       ;TEST NUMBER
004520  012737  004542  001106                MOV     #TEST2,$LPADR   ;LOAD LOOP ON TEST ADDRESS
004526  012737  004542  001110                MOV     #TEST2,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
004534  012737  000012  001176                MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
004542  012706  001100            TEST2:      MOV     #STACK,SP       ;LOAD THE STACK POINTER
004546  012737  000240  177776                MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
004554  005037  001256                        CLR     TIMEA           ;CLEAR TIMEOUT VALUE FOR PORT A
004560  005037  001260                        CLR     TIMEAP  ;CLEAR UPPER TIMEOUT TOLERANCE
004564  005037  001262                        CLR     TIMEAM  ;CLEAR LOWER TIMEOUT TOLERANCE

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

004570  005037  001252                        CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
004574  012737  003720  001254                MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS

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

                                        ;SEIZE THE DRIVE THROUGH PORT A

004602  113760  001224  000010                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
004610  013737  001224  001236                MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
004616  005060  000012                        CLR     RPDS1(R0)       ;WRITE RPDS1
004622  113760  001226  000010                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
004630  013737  001226  001234                MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004636  013737  001226  001240                MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
004644  016037  000012  001126                MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
004652  010037  001122                        MOV     R0,$BDADR       ;RH11 BASE ADDRESS
004656  062737  000012  001122                ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
004664  005037  001124                        CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
004670  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
004676  001403                                BEQ     64$             ;BR IF IT IS
004700  104004                                ERROR   4               ;REPORT THE ERROR
004702  000137  006066                        JMP     5$              ;BYPASS REST OF THE SUBTEST
004706                                  64$:
004706  113760  001224  000010                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
004714  013737  001224  001234                MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004722  016037  000012  001126                MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
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,$BDADR       ;BASE ADDRESS F R REGISTER RPDT
2477  005606  062737  000026  001122  ADD     #RPDT,$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$:     SCOPE                   ;LOOP ?
 2527
 2528                                  ;;*****************************************************************
 2529                                  ;*TEST 3         PORT 'B' SEIZE/TIMEOUT TEST
 2530                                  ;*
 2531                                  ;*VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
 2532                                  ;*      THE DRIVE.
 2533                                  ;*
 2534                                  ;*  A.  WRITE 0'S INTO RPDS1 THROUGH PORT 'B';  VERIFY THAT THE DRIVE
 2535                                  ;*      HAS BEEN SEIZED.
 2536                                  ;*
 2537                                  ;*  B.  READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'A';
 2538                                  ;*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
 2539                                  ;*
 2540                                  ;*  C.  WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
 2541                                  ;*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
 2542                                  ;*      VALUE FOR LATER USE.  VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
 2543                                  ;*      NEUTRAL.
 2544                                  ;*
 2545                                  ;;*****************************************************************
 2546  006070                          TST3:
 2547  006070  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 2548  006074  001406                          BEQ     2$              ;BR IF NOT
 2549  006076  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 2550  006100  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 2551  006104  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 2552  006112  112737  000003  001102  2$:     MOVB    #3,$TSTNM       ;TEST NUMBER
 2553  006120  012737  006142  001106          MOV     #TEST3,$LPADR   ;LOAD LOOP ON TEST ADDRESS
 2554  006126  012737  006142  001110          MOV     #TEST3,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
 2555  006134  012737  000012  001176          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 2556  006142  012706  001100          TEST3:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 2557  006146  012737  000240  177776          MOV     #<5*32.>,@#PS   ;SET PRIORITY TO 5 IN CASE LOOPING
 2558  006154  005037  001264                  CLR     TIMEB           ;CLEAR TIMEOUT VALUE FOR PORT B
 2559  006160  005037  001266                  CLR     TIMEBP  ;CLEAR UPPER TIMEOUT TOLERANCE
 2560  006164  005037  001270                  CLR     TIMEBM  ;CLEAR LOWER TIMEOUT TOLERANCE
 2561
 2562                                  ;;*****************************************************************
 2563                                  ;START THE TIMER
 2564
 2565  006170  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
 2566  006174  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
 2567
 2568                                  ;;*****************************************************************
 2569
 2570                                          ;SEIZE THE DRIVE THROUGH PORT B
 2571
 2572  006202  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 2573  006210  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 2574  006216  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
 2575  006222  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 2576  006230  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 2577  006236  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
```

```
2578  006244  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;SEE IF DRIVE SEIZED BY PORT B
2579  006252  010037  001122                  MOV     R0,$BDADR         ;RH11 BASE ADDRESS
2580  006256  062737  000012  001122          ADD     #RPDS1,$BDADR     ;GENERATE BAD REGISTER ADDRESS
2581  006264  005037  001124                  CLR     $GDDAT            ;REGISTER SHOULD BE ZERO
2582  006270  023737  001124  001126          CMP     $GDDAT,$BDDAT     ;IS THE REGISTER ZERO
2583  006276  001403                          BEQ     64$               ;BR IF IT IS
2584  006300  104004                          ERROR   4                 ;REPORT THE ERROR
2585  006302  000137  007466                  JMP     5$                ;BYPASS REST OF THE SUBTEST
2586  006306                          64$:
2587  006306  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B
2588  006314  013737  001226  001234          MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2589  006322  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
2590  006330  012737  011600  001124          MOV     #MOL!PGM!DPR!DRY,$GDDAT   ;EXPECTED STATUS
2591  006336  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
2592  006344  005137  001166                  COM     $TMP1             ;COMPLEMENT THE EXPECTED STATUS
2593  006350  013737  001126  001164          MOV     $BDDAT,$TMP0      ;SAVE THE ACTUAL STATUS
2594  006356  043737  001166  001164          BIC     $TMP1,$TMP0       ;CLEAR UNWANTED BITS
2595  006364  023737  001124  001164          CMP     $GDDAT,$TMP0      ;ARE THE EXPECTED STATUS BITS SET ?
2596  006372  001401                          BEQ     65$               ;BR IF THEY ARE
2597  006374  104005                          ERROR 5                   ;REPORT THE ERROR
2598  006376  000240                  65$:    NOP
2599
2600                                  ;;*****************************************************************
2601                                  ;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK
2602
2603  006400  113760  001224  000010          MOVB    PORTA,RPCS2(R0)   ;SELECT PORT A
2604  006406  013737  001224  001234          MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2605  006414  016046  000046                  MOV     RPEC2(R0),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
2606  006420  016046  000044                  MOV     RPEC1(R0),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
2607  006424  016046  000042                  MOV     RPER3(R0),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
2608  006430  016046  000030                  MOV     RPSN(R0),-(SP)  ;STORE REGISTER RPSN, PORT A, FOR CHECK
2609  006434  016046  000036                  MOV     RPCC(R0),-(SP)  ;STORE REGISTER RPCC, PORT A, FOR CHECK
2610  006440  016046  000034                  MOV     RPCA(R0),-(SP)  ;STORE REGISTER RPCA, PORT A, FOR CHECK
2611  006444  016046  000032                  MOV     RPOF(R0),-(SP)  ;STORE REGISTER RPOF, PORT A, FOR CHECK
2612  006450  016046  000040                  MOV     RPER2(R0),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
2613  006454  016046  000020                  MOV     RPLA(R0),-(SP)  ;STORE REGISTER RPLA, PORT A, FOR CHECK
2614  006460  016046  000026                  MOV     RPDT(R0),-(SP)  ;STORE REGISTER RPDT, PORT A, FOR CHECK
2615  006464  016046  000006                  MOV     RPDA(R0),-(SP)  ;STORE REGISTER RPDA, PORT A, FOR CHECK
2616  006470  016046  000024                  MOV     RPMR(R0),-(SP)  ;STORE REGISTER RPMR, PORT A, FOR CHECK
2617  006474  016046  000014                  MOV     RPER1(R0),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK
2618
2619                                  ;;*****************************************************************
2620                                  ;WAIT FOR PORT B TO TIMEOUT
2621
2622  006500  005760  000012          1$:     TST     RPDS1(R0)         ;WAIT FOR THE DRIVE TO TIMEOUT
2623  006504  001006                          BNE     2$                ;BR WHEN TIMEOUT OCCURS
2624  006506  005737  001254                  TST     WATCH             ;CHECK WATCH
2625  006512  001372                          BNE     1$                ;BR IF NOT ZERO
2626  006514  104036                          ERROR   36                ;NO TIMEOUT WITHIN 2 SECONDS
2627  006516  000137  007106                  JMP     4$                ;BYPASS TIMEOUT TIME CHECK
2628  006522  012737  000340  177776  2$:     MOV     #<7*32.>,@#PS     ;SET PRIORITY TO 7 TO STOP CLOCK
2629  006530  013737  001252  001264          MOV     TIME,TIMEB        ;SAVE THE ELAPSED TIME FOR PORT B
2630  006536  004537  056630                  JSR     R5,TOLER          ;CALCULATE THE TOLERANCE
2631  006542  001264                          .WORD   TIMEB             ;TIMEOUT VALUE FOR PORT B
2632  006544  012637  001266                  MOV     (SP)+,TIMEBP      ;+25% TOLERANCE
2633  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,$GDDAT  ;COMPARISON CONSTANT
006620  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
006626  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
006634  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
006642  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006650  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
006656  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
006664  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
006672  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
006700  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
006706  001006                          BNE     66$             ;BR IF NOT
006710  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
006714  001037                          BNE     68$             ;BR IF NOT
006716  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
006720  000137  007104                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
006724  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
006732  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006740  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
006746  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
006752  001414                          BEQ     67$             ;BR IF ZERO
006754  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
006762  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
006770  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
006776  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
007002  001004                          BNE     68$             ;BR IF NOT
007004  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
007012  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
007014  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
007022  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
007030  042737  100100  001170          BIC     #ATA!VV,$TMP2   ;DON'T CHECK ATTN BIT OR VV BIT
007036  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
007044  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
007046  104007                          ERROR   7               ;REPORT ERROR
007050  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
007056  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
007064  042737  100100  001172          BIC     #ATA!VV,$TMP3   ;DON'T CHECK ATTN BIT OR VV BIT
007072  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
007100  001401                          BEQ     70$             ;BR IF Ok
```

```
                                ERROR   7               ;REPORT ERROR
007104  000240          70$:    NOP
                        ;;*****************************************************************
                        ;CHECK THE REGISTERS STORED THROUGH PORT A.  ALL REGISTERS SHOULD BE ZERO.
                        ;THE REGISTERS ARE STORED ON THE STACK.
007106  013737  001224  001234  4$:     MOV     PORTA,PTNBR     ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007114  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER1
007120  062737  000014  001122  ADD     #RPER1,$BDADR   ;ADDRESS OF RPER1 FOR TYPEOUT
007126  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER1
007132  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007134  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007136  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPMR
007142  062737  000024  001122  ADD     #RPMR,$BDADR    ;ADDRESS OF RPMR FOR TYPEOUT
007150  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPMR
007154  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007156  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007160  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDA
007164  062737  000006  001122  ADD     #RPDA,$BDADR    ;ADDRESS OF RPDA FOR TYPEOUT
007172  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDA
007176  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007200  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007202  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPDT
007206  062737  000026  001122  ADD     #RPDT,$BDADR    ;ADDRESS OF RPDT FOR TYPEOUT
007214  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPDT
007220  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007222  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007224  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPLA
007230  062737  000020  001122  ADD     #RPLA,$BDADR    ;ADDRESS OF RPLA FOR TYPEOUT
007236  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPLA
007242  001401                  BEQ     .+4             ;CONTENTS ZERO ?
 07244  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007246  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPER2
007252  062737  000040  001122  ADD     #RPER2,$BDADR   ;ADDRESS OF RPER2 FOR TYPEOUT
007260  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPER2
007264  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007266  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007270  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPOF
007274  062737  000032  001122  ADD     #RPOF,$BDADR    ;ADDRESS OF RPOF FOR TYPEOUT
007302  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPOF
007306  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007310  104006                  ERROR   6               ;REPORT THAT PORT A SAW NON-ZERO REGISTER
007312  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCA
007316  062737  000034  001122  ADD     #RPCA,$BDADR    ;ADDRESS OF RPCA FOR TYPEOUT
007324  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCA
007330  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007332  104006                  ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
007334  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPCC
007340  062737  000036  001122  ADD     #RPCC,$BDADR    ;ADDRESS OF RPCC FOR TYPEOUT
007346  012637  001126          MOV     (SP)+,$BDDAT    ;CHECK THE STORED CONTENTS OF RPCC
007352  001401                  BEQ     .+4             ;CONTENTS ZERO ?
007354  104006                  ERROR   6               ;REPORT THAT PORT A SEES NON-ZERO REGISTER
007356  010037  001122          MOV     R0,$BDADR       ;BASE ADDRESS FOR REGISTER RPSN
007362  062737  000030  001122  ADD     #RPSN,$BDADR    ;ADDRESS OF RPSN FOR TYPEOUT
007370  012637  001126          MOV     (SP)+,$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
2765
2766                                  ;*****************************************************************
2767                                  ;*TEST 4        PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
2768                                  ;*
2769                                  ;*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED.  SET 'VV'
2770                                  ;*      FOR THE PORT UNDER TEST.
2771                                  ;*
2772                                  ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
2773                                  ;*      DRIVE WAS SEIZIED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
2774                                  ;*
2775                                  ;*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'.  VERIFY THAT THE
2776                                  ;*      'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
2777                                  ;*      FOR PORT 'B'.  (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
2778                                  ;*      'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
2779                                  ;*
2780                                  ;*  C.  STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
2781                                  ;*      THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
2782                                  ;*
2783                                  ;*****************************************************************
2784  007470                  TST4:
2785  007470  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
2786  007474  001406                          BEQ     2$              ;BR IF NOT
2787  007476  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
2788  007500  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
2789  007504  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
2790  007512  112737  000004  001102  2$:     MOVB    #4,$TSTNM       ;TEST NUMBER
2791  007520  012737  007542  001106          MOV     #TEST4,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2792  007526  012737  007542  001110          MOV     #TEST4,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
2793  007534  012737  000001  001176          MOV     #1,$TIMES       ;;DO 1 ITERATION
2794  007542  012706  001100          TEST4:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2795  007546  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
2796  007554  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2797
2798                                  ;*****************************************************************
2799                                  ;START THE TIMER
2800
2801  007562  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
```

```
2802  007566  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
2803  007574  013737  001224  001236          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
2804
2805                                          ;;*****************************************************************
2806                                          ;ISSUE DRIVE CLEAR COMMAND
2807
2808  007602  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
2809
2810                                          ;;*****************************************************************
2811                                          ;VERIFY THAT DRIVE SEIZED BY PORT A.
2812
2813  007610  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
2814  007616  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2815  007624  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
2816  007630  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
2817  007636  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2818  007644  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
2819  007650  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
2820  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
```

```
                                        ;;*****************************************************************
                                        ;ISSUE READIN PRESET COMMAND AND SET FMT22

010130  012760  000023  000000                MOV     #23,RPCS1(R0)   ;ISSUE A READIN PRESET
010136  012760  010000  000032                MOV     #FMT22,RPOF(R0) ;SET FMT22

                                        ;;*****************************************************************
                                        ;VERIFY THAT THE DRIVE STATUS IS CORRECT

010144  005037  001244                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010150  016037  000012  001126                MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
010156  012737  000012  001122                MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010164  060037  001122                        ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
010170  012737  011700  001124                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
010176  013737  001126  001164                MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010204  042737  106077  001164                BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
010212  023737  001124  001164                CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
010220  001414                                BEQ     70$             ;BR IF OK
010222  013737  001126  001174                MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
010230  042737  071700  001174                BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
010236  053737  001174  001124                BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
010244  104013                                ERROR   13              ;TYPE MESSAGE 13
010246  005137  001244                        COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
010252  000240                        70$:    NOP
010254  113760  001226  000010                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
010262  013737  001226  001234                MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

010270  005760  000012                1$:     TST     RPDS1(R0)       ;WAIT FOR THE PORT TO TIME OUT
010274  001006                                BNE     2$              ;BR WHEN TIMEOUT OCCURS
010276  005737  001254                        TST     WATCH           ;CHECK THE WATCH
010302  001372                                BNE     1$              ;BR IF NOT ZERO
010304  104036                                ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
010306  000137  010624                        JMP     3$              ;BYPASS ATTN REGISTER CHECK

                                        ;;*****************************************************************
                                        ;SEE IF DRIVE RETURNED TO NEUTRAL

010312                                  2$:

                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

010312  005037  001250                        CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
010316  012737  000012  001122                MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
010324  060037  001122                        ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
010330  012737  011600  001124                MOV     #MOL!PGM!DPR!DRY,$GDDAT  ;COMPARISON CONSTANT
010336  113760  001224  000010                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
010344  016037  000012  001170                MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
010352  013737  001170  001164                MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
010360  042737  100100  001164                BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010366  113760  001226  000010                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
010374  016037  000012  001172                MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
010402  013737  001172  001166                MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
010410  042737  100100  001166                BIC     #ATA!VV,$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 COMMANC 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                          5$:
 2964  010626  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 2965  010632  001406                          BEQ     2$              ;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,$TIMES       ;;DO 1 ITERATION
 2973  010700  012706  001100          TEST5:  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 2974  010704  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 2975  010712  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 2976
 2977                                  ;;*****************************************************************
 2978                                  ;START THE TIMER
 2979
 2980  010720  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
 2981  010724  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
 2982  010732  013737  001226  001236          MOV     PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS
 2983
 2984                                  ;;*****************************************************************
 2985                                  ;ISSUE DRIVE CLEAR COMMAND
 2986
 2987  010740  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
 2988
 2989                                  ;;*****************************************************************
 2990                                  ;VERIFY THAT DRIVE SEIZED BY PORT B.
 2991
 2992  010746  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 2993  010754  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 2994  010762  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 2995  010766  016037  000012  001126          MOV     RPDS1(R0),$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                                  ;*
```

```
                                        ;;*****************************************************************
                                        ;TST6:
011764                                  
011764  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
011770  001406                          BEQ     2$              ;BR IF NOT
011772  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
011774  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
012000  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
012006  112737  000006  001102  2$:     MOVB    #6,$TSTNM       ;TEST NUMBER
012014  012737  012036  001106          MOV     #TEST6,$LPADR   ;LOAD LOOP ON TEST ADDRESS
012022  012737  012036  001110          MOV     #TEST6,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
012030  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
012036  012706  001100          TEST6:  MOV     #STACK,SP       ;LOAD THE STACK POINTER

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

012042  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
012046  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS

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

                                        ;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     ??$             ;BYPASS THE REST OF THE CHECKS
```

```
3194  012254  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
3195  012262  013737  001226  001234          MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3196  012270  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
3197  012276  005737  001164                  TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
3198  012302  001414                          BEQ     67$              ;BR IF ZERO
3199  012304  013737  001224  001234          MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
3200  012312  013737  001172  001126          MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
3201  012320  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
3202  012326  005737  001166                  TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
3203  012332  001004                          BNE     68$              ;BR IF NOT
3204  012334  012737  177777  001250  67$:    MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
3205  012342  104026                          ERROR   26               ;TYPE ERROR MESSAGE 26
3206  012344  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
3207  012352  013737  001224  001234          MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
3208  012360  042737  100000  001170          BIC     #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
3209  012366  023737  001124  001170          CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
3210  012374  001401                          BEQ     69$              ;BR IF OK FROM PORT A.
3211  012376  104007                          ERROR   7                ;REPORT ERROR
3212  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
3269  012750  104036                          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
3270  012752  000004                  2$:     SCOPE                   ;LOOP ?
3271
3272                                          ;;*****************************************************************
3273                                          ;*TEST 7        TEST RELEASE, DRIVE SEIZED BY PORT 'B'
3274                                          ;*
3275                                          ;*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
3276                                          ;*
3277                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
3278                                          ;*
3279                                          ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
3280                                          ;*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
3281                                          ;*      DRIVE.
3282                                          ;*
3283                                          ;;*****************************************************************
3284  012754                          TST7:
3285  012754  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3286  012760  001406                          BEQ     2$              ;BR IF NOT
3287  012762  100002                          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
```

```
                                        ;;*****************************************************************
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
013744                                  TST10:
013744  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
013750  001406                                  BEQ     2$              ;BR IF NOT
013752  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
013754  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
013760  012737  177777  001274    1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
013766  112737  000010  001102    2$:     MOVB    #10,$TSTNM      ;TEST NUMBER
013774  012737  014016  001106            MOV     #TEST10,$LPADR  ;LOAD LOOP ON TEST ADDRESS
014002  012737  014016  001110            MOV     #TEST10,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
014010  012737  000144  001176            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
014016  012706  001100                TEST10: MOV     #STACK,SP       ;LOAD THE STACK POINTER
014022  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
014030  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014036  013737  001224  001236            MOV     PORTA,SEIZPT    ;ADDR OF PORT WHICH WILL ISSUE RELEASE

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

014044  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND

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


                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL

014052  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
014056  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
014064  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
014070  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
014076  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
014104  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
014112  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
014120  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014126  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
014134  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
014142  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
014150  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014156  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
014164  001006                          BNE     64$             ;BR IF NOT
014166  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
014172  001037                          BNE     66$             ;BR IF NOT
014174  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
014176  000137  014362                  JMP     68$             ;BYPASS THE REST OF THE CHECKS
014202  013737  001170  001126  64$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014210  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
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$:     NOP
3496  014364  000004                                SCOPE                   ;LOOP ?
3497
3498                                        ;;*****************************************************************
3499                                        ;*TEST 11       TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
3500                                        ;*
3501                                        ;*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
3502                                        ;*
3503                                        ;*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
3504                                        ;*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
3505                                        ;*
3506                                        ;;*****************************************************************
3507  014366                                T11:
3508  014366  005737  001274                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
3509  014372  001406                                BEQ     2$              ;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     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
3548  014614  001007                          BNE     66$             ;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'.
```

```
                                                T12     TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'A'                  SEQ 0067

                                                        ;*      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.
                                                        ;*
                                                        ;;*****************************************************************
015010                                          TST12:
015010  005737  001274                                  TST     KYBCTL                  ;PERFORMING ONLY SINGLE TESTS ?
015014  001406                                          BEQ     2$                      ;BR IF NOT
015016  100002                                          BPL     1$                      ;BR IF JUST ENTERED TEST
015020  000137  002602                                  JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
015024  012737  177777  001274              1$:       MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
015032  112737  000012  001102              2$:       MOVB    #12.,$TSTNM             ;TEST NUMBER
015040  012737  015062  001106                        MOV     #TEST12,$LPADR          ;LOAD LOOP ON TEST ADDRESS
015046  012737  015062  001110                        MOV     #TEST12,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
015054  012737  007640  001176                        MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS
015062  012706  001100                      TEST12:   MOV     #STACK,SP               ;LOAD THE STACK POINTER

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

                                                        ;SEIZE THE DRIVE THROUGH PORT A

015066  113760  001224  000010                        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
015074  013737  001224  001236                        MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
015102  005060  000012                              CLR     RPDS1(R0)       ;WRITE RPDS1
015106  113760  001226  000010                      MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
015114  013737  001226  001234                      MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015122  013737  001226  001240                      MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
015130  016037  000012  001126                      MOV     RPDS1(R0),$BDDAT        ;SEE IF DRIVE SEIZED BY PORT A
015136  010037  001122                              MOV     R0,$BDADR       ;RH11 BASE ADDRESS
015142  062737  000012  001122                      ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
015150  005037  001124                              CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
015154  023737  001124  001126                      CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
015162  001403                                      BEQ     64$             ;BR IF IT IS
015164  104004                                      ERROR   4               ;REPORT THE ERROR
015166  000137  016364                              JMP     1$              ;BYPASS REST OF THE SUBTEST
015172                                      64$:
015172  113760  001224  000010                      MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
015200  013737  001224  001234                      MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015206  016037  000012  001126                      MOV     RPDS1(R0),$BDDAT        ;SEE IF SEIZING PORT SEES CORRECT STATUS
015214  012737  011700  001124                      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT      ;EXPECTED STATUS
015222  013737  001124  001166                      MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
015230  005137  001166                              COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
015234  013737  001126  001164                      MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
015242  043737  001166  001164                      BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
015250  023737  001124  001164                      CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
015256  001401                                      BEQ     65$             ;BR IF THEY ARE
015260  104005                                      ERROR 5                 ;REPORT THE ERROR
015262  000240                              65$:    NOP
```

```
                                        ;;*****************************************************************
                                        ;DRIVE CLEAR THROUGH PORT A FIRST

015264  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT A

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

015272  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
015300  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015306  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
015312  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
015320  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015326  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
015332  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
015336  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
015344  042737  100000  001164          BIC     #^C77777,$TMP0   ;SAVE SPECIFIED BITS
015352  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
015360  001414                          BEQ     66$              ;BR IF OK
015362  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
015370  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
015376  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
015404  104033                          ERROR   33               ;TYPE MESSAGE 33
015406  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
015412  000240                  66$:    NOP
015414  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
015422  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015430  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
015434  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
015442  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015450  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
015454  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
015462  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
015470  042737  100000  001164          BIC     #^C77777,$TMP0   ;SAVE SPECIFIED BITS
015476  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
015504  001414                          BEQ     68$              ;BR IF OK
015506  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
015514  042737  077777  001174          BIC     #77777,$TMP4     ;CLEAR THE MASKED BITS
015522  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
015530  104033                          ERROR   33               ;TYPE MESSAGE 33
015532  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
015536  000240                  68$:    NOP

                                        ;;*****************************************************************
                                        ;NOW ISSUE MASSBUS INIT

015540  012760  000040  000010          MOV     #CLR,RPCS2(R0)  ;ISSUE MASSBUS INIT

                                        ;;*****************************************************************
                                        ;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

015546  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
015554  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015562  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
015566  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
015574  012737  000012  001122          MOV     #RPDS1,$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.
3773  016324  104007                          ERROR   7                 ;REPORT ERROR
3774  016326  013737  001172  001126  77$:    MOV     $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
3775  016334  013737  001226  001234          MOV     PORTB,PTNBR       ;CHANGE PORT NUMBER
3776  016342  042737  100000  001172          BIC     #ATA,$TMP3        ;DON'T CHECK THE ATTN BIT
3777  016350  023737  001124  001172          CMP     $GDDAT,$TMP3      ;SEE IF READ OK FROM PORT B.
3778  016356  001401                          BEQ     78$               ;BR IF OK
3779  016360  104007                          ERROR   7                 ;REPORT ERROR
3780  016362  000240                  78$:    NOP
3781  016364  000004                  1$:     SCOPE                     ;LOOP ?
3782
3783                                          ;;**********************************************************************
3784                                          ;*TEST 13       TEST THAT 'CLEAR' DOES NOT CAUSE RELEASE FROM PORT 'B'
3785                                          ;*
3786                                          ;*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
3787                                          ;*      PORT TO RELEASE THE DRIVE.
3788                                          ;*
3789                                          ;*  A.  SEIZE THE DRIVE BY WRITING 0'S INTO RPDS1 THROUGH PORT 'B'.
3790                                          ;*      VERIFY THAT THE DRIVE HAS BEEN SEIZED.
3791                                          ;*
3792                                          ;*  B.  ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
3793                                          ;*      DOES NOT RETURN TO NEUTRAL.
3794                                          ;*
3795                                          ;*  C.  ISSUE A MASSBUS CLEAR THROUGH THE RH11 AND VERIFY THAT THE DRIVE
3796                                          ;*      DOES NOT RETURN TO NEUTRAL.
3797                                          ;*
3798                                          ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
3799                                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
3800                                          ;*
3801                                          ;;**********************************************************************
3802  016366                          TST13:
3803  016366  005737  001274                  TST     KYBCTL            ;PERFORMING ONLY SINGLE TESTS ?
3804  016372  001406                          BEQ     2$                ;BR IF NOT
3805  016374  100002                          BPL     1$                ;BR IF JUST ENTERED TEST
3806  016376  000137  002602                  JMP     EXEC              ;RETURN & GET NEXT TEST NUMBER
3807  016402  012737  177777  001274  1$:     MOV     #-1,KYBCTL        ;SET SINGLE TEST INDICATOR
3808  016410  112737  000013  001102  2$:     MOVB    #13,$TSTNM        ;TEST NUMBER
3809  016416  012737  016440  001106          MOV     #TEST13,$LPADR    ;LOAD LOOP ON TEST ADDRESS
```

```
3810  016424  012737  016440  001110          MOV     #TEST13,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3811  016432  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
3812  016440  012706  001100          TEST13: MOV     #STACK,SP       ;LOAD THE STACK POINTER
3813
3814                                  ;;**************************************************************
3815
3816                                  ;SEIZE THE DRIVE THROUGH PORT B
3817
3818  016444  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
3819  016452  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
3820  016460  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
3821  016464  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
3822  016472  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3823  016500  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
3824  016506  016037  000012  001126          MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
3825  016514  010037  001122                  MOV     R0,$BDADR       ;RH11 BASE ADDRESS
3826  016520  062737  000012  001122          ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
3827  016526  005037  001124                  CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
3828  016532  023737  001124  001126          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!DRV!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                                          ;*
 3997                                          ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 3998                                          ;*
 3999                                          ;*  C.  ISSUE A MASSBUS CLEAR.
 4000                                          ;*
 4001                                          ;*
 4002                                          ;*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE ATTENTION
 4003                                          ;*      BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
 4004                                          ;*      'B' IS STILL SET.
 4005                                          ;*
 4006                                          ;*****************************************************************
 4007  017744                                  TST14:
 4008  017744  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 4009  017750  001406                          BEQ     2$              ;BR IF NOT
 4010  017752  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 4011  017754  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 4012  017760  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 4013  017766  112737  000014  001102  2$:     MOVB    #14,$TSTNM      ;TEST NUMBER
 4014  017774  012737  020016  001106          MOV     #TEST14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4015  020002  012737  020016  001110          MOV     #TEST14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4016  020010  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 4017  020016  012706  001100          TEST14: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4018                                          ;;*****************************************************************
 4019
 4020                                          ;SET ATTENTION BITS FOR BOTH PORTS
 4021
 4022  020022  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT 64$
 4023  020030  012760  177777  000014          MOV     #-1,RPER1(R0)   ;FORCE ERRORS
 4024  020036  005060  000014                  CLR     RPER1(R0)       ;CLEAR THE ERRORS
 4025  020042  013760  001226  000010          MOV     PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
 4026  020050  005760  000012          64$:    TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
 4027  020054  001775                          BEQ     64$             ;BR IF DRIVE HASN'T TIMED OUT
 4028  020056  012760  177777  000014          MOV     #-1,RPER1(R0)   ;FORCE ERRORS ON PORT 65$
 4029  020064  005060  000014                  CLR     RPER1(R0)       ;CLEAR THE ERRORS
 4030  020070  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT "64$" AGAIN
 4031  020076  005760  000012          65$:    TST     RPDS1(R0)       ;WAIT FOR DRIVE TO TIMEOUT
 4032  020102  001775                          BEQ     65$             ;BR IF DRIVE HASN'T TIMED OUT
 4033
```

```
                                ;;*****************************************************************
                                ;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 TYPECUT
020220  104010                          ERROR   10               ;REPORT THE ERROR
020222  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
020226  000240                  66$:    NOP
020230  005737  001244                  TST     CKERR            ;WAS ATTN BIT FOR PORT A SET ?
020234  001402                          BEQ     .+6              ;BR IF IT WAS
020236  000137  021244                  JMP     1$               ;BYPASS REST OF TEST IF NOT
020242  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
020250  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020256  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
020262  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
020270  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020276  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
020302  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
020310  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
020316  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
020324  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
020332  001414                          BEQ     68$              ;BR IF OK
020334  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
020342  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
020350  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
020356  104010                          ERROR   10               ;REPORT THE ERROR
020360  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
020364  000240                  68$:    NOP
020366  005737  001244                  TST     CKERR            ;WAS ATTN BIT FOR PORT B SET ?
020372  001402                          BEQ     .+6              ;BR IF IT WAS
020374  000137  021244                  JMP     1$               ;BYPASS REST OF TEST IF NOT

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

                                        ;SEIZE THE DRIVE THROUGH PORT A

020400  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
020406  013737  001224  001236          MOV     PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
020414  005060  000012                  CLR     RPDS1(R0)        ;WRITE RPDS1
020420  013737  001226  001240          MOV     PORTB,OPPRT      ;'OPPOSITE' PORT ADDRESS

                                ;;*****************************************************************
                                ;ISSUE MASSBUS INIT TO PORT A
```

```
 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 TYPECUT
 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.
```

```
                                        ;*
                                        ;*  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.
                                        ;*
                                        ;;*****************************************************************
021246                                  TST15:
021246  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
021252  001406                                  BEQ     2$              ;BR IF NOT
021254  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
021256  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
021262  012737  177777  001274    1$:           MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
021270  112737  000015  001102    2$:           MOVB    #15,$TSTNM      ;TEST NUMBER
021276  012737  021320  001106                  MOV     #TEST15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
021304  012737  021320  001110                  MOV     #TEST15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
021312  012737  000004  001176                  MOV     #4,$TIMES       ;;DO 4 ITERATIONS
021320  012706  001100            TEST15:       MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                        ;;*****************************************************************

                                                ;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
```

```
 4258  021544  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 4259  021552  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 4260  021560  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
 4261  021564  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 4262  021572  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 4263  021600  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
 4264  021604  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
 4265  021612  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
 4266  021620  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
 4267  021626  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
 4268  021634  001414                          BEQ     68$              ;BR IF OK
 4269  021636  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
 4270  021644  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
 4271  021652  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
 4272  021660  104010                          ERROR   10               ;REPORT THE ERROR
 4273  021662  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
 4274  021666  000240                  68$:    NOP
 4275  021670  005737  001244                  TST     CKERR            ;WAS ATTN BIT FOR PORT A SET ?
 4276  021674  001402                          BEQ     .+6              ;BR IF IT WAS
 4277  021676  000137  022546                  JMP     1$               ;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                                  ;;*****************************************************************
```

```
                                                ;RELEASE THE DRIVE FROM PORT B

022052  113760  001226  000010                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
022060  013737  001226  001234                MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022066  012760  000013  000000                MOV     #13,RPCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

022074  005037  001250                        CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
022100  012737  000012  001122                MOV     #RPDS1,$BDADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
022106  060037  001122                        ADD     R0,$BDADR        ;ADD THE I/O BASE ADDRESS
022112  012737  011700  001124                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
022120  113760  001224  000010                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
022126  016037  000012  001170                MOV     RPDS1(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022134  013737  001170  001164                MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
022142  042737  100100  001164                BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022150  113760  001226  000010                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
022156  016037  000012  001172                MOV     RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022164  013737  001172  001166                MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
022172  042737  100100  001166                BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022200  023737  001164  001166                CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
022206  001006                                BNE     74$              ;BR IF NOT
022210  005737  001164                        TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
022214  001045                                BNE     76$              ;BR IF NOT
022216  104046                                ERROR   46               ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
022220  000137  022420                        JMP     78$              ;BYPASS THE REST OF THE CHECKS
022224  013737  001170  001126  74$:          MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
022232  013737  001226  001234                MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
022240  113760  001226  000010                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
022246  005737  001164                        TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
022252  001414                                BEQ     75$              ;BR IF ZERO
022254  013737  001224  001234                MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
022262  013737  001172  001126                MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
022270  113760  001224  000010                MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
022276  005737  001166                        TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
022302  001012                                BNE     76$              ;BR IF NOT
022304  012737  177777  001250  75$:          MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
022312  012760  000011  000000                MOV     #11,RPCS1(R0)    ;CLEAR THE DRIVE
022320  012760  000013  000000                MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
022326  104026                                ERROR   26               ;TYPE ERROR MESSAGE 26
022330  013737  001170  001126  76$:          MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
022336  013737  001224  001234                MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
022344  042737  100000  001170                BIC     #ATA,$TMP2       ;DON'T CHECK THE ATTN BIT
022352  023737  001124  001170                CMP     $GDDAT,$TMP2     ;ALL BITS OK ?
022360  001401                                BEQ     77$              ;BR IF OK FROM PORT A.
022362  104007                                ERROR   7                ;REPORT ERROR
022364  013737  001172  001126  77$:          MOV     $TMP3,$BDDAT     ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
022372  013737  001226  001234                MOV     PORTB,PTNBR      ;CHANGE PORT NUMBER
022400  042737  100000  001172                BIC     #ATA,$TMP3       ;DON'T CHECK THE ATTN BIT
022406  023737  001124  001172                CMP     $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
022414  001401                                BEQ     78$              ;BR IF OK
022416  104007                                ERROR   7                ;REPORT ERROR
022420  000240                        78$:     NOP

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

```
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 TYPEOUT
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
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585  024010                          ;*********************************************************************
4586  024010  005737  001274          ;*TST17:
4587  024014  001406                  ;*      TST     KYBCTL              ;PERFORMING ONLY SINGLE TESTS ?
4588  024016  100002                  ;*      BEQ     2$                  ;BR IF NOT
4589  024020  000137  002602          ;*      BPL     1$                  ;BR IF JUST ENTERED TEST
4590  024024  012737  177777  001274 ;*      JMP     EXEC                ;RETURN & GET NEXT TEST NUMBER
4591  024032  112737  000017  001102 ;*1$:   MOV     #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
4592  024040  012737  024062  001106 ;*2$:   MOVB    #17,$TSTNM          ;TEST NUMBER
4593  024046  012737  024062  001110 ;*      MOV     #TEST17,$LPADR      ;LOAD LOOP ON TEST ADDRESS
                                      ;*      MOV     #TEST17,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
```

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

```
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                                    ;*
```

```
                                        ;;*****************************************************************
024676                                  T$T20:
024676  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
024702  001406                                  BEQ     2$              ;BR IF NOT
024704  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
024706  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
024712  012737  177777  001274          1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
024720  112737  000020  001102          2$:     MOVB    #20,$TSTNM      ;TEST NUMBER
024726  012737  024750  001106                  MOV     #TEST20,$LPADR  ;LOAD LOOP ON TEST ADDRESS
024734  012737  024750  001110                  MOV     #TEST20,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
024742  012737  007640  001176                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
024750  012706  001100          TEST20: MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS
024754  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
024762  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
024766  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
024774  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
025002  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
025010  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
025014  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
025022  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                        ;;*****************************************************************
                                        ;SEIZE THE DRIVE THROUGH PORT B
025030  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
025036  013737  001226  001236                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
025044  005760  000000                          TST     RPCS1(R0)       ;READ RHCS1
025050  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
025056  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025064  013737  001224  001240                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
025072  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
025100  010037  001122                          MOV     R0,$BDADR       ;RH11 BASE ADDRESS
025104  062737  000012  001122                  ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
025112  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
025116  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
025124  001403                                  BEQ     64$             ;BR IF IT IS
025126  104004                                  ERROR   4               ;REPORT THE ERROR
025130  000137  025562                          JMP     1$              ;BYPASS REST OF THE SUBTEST
025134                                  64$:
025134  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
025142  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025150  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
025156  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
025164  013737  001124  001166                  MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
025172  005137  001166                          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
025176  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
025204  043737  001166  001164                  BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
025212  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
025220  001401                                  BEQ     65$             ;BR IF THEY ARE
025222  104005                                  ERROR 5                 ;REPORT THE ERROR
025224  000240                          65$:    NOP
```

```
                                        ;;*****************************************************************
                                                ;RELEASE THE DRIVE FROM PORT B
                                                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                                MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

                                                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
                                                MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
                                                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
                                                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT   ;COMPARISON CONSTANT
                                                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
                                                MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
                                                MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
                                                BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
                                                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
                                                MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
                                                MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
                                                BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
                                                CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
                                                BNE     66$             ;BR IF NOT
                                                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
                                                BNE     68$             ;BR IF NOT
                                                ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
                                                JMP     70$             ;BYPASS THE REST OF THE CHECKS
                                        66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
                                                MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
                                                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
                                                BEQ     67$             ;BR IF ZERO
                                                MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                                MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
                                                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
                                                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
                                                BNE     68$             ;BR IF NOT
                                        67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
                                                MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
                                                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                                ERROR   26              ;TYPE ERROR MESSAGE 26
                                        68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
                                                MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
                                                CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
                                                BEQ     69$             ;BR IF OK FROM PORT A.
                                                ERROR   7               ;REPORT ERROR
                                        69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                                MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
                                                CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
                                                BEQ     70$             ;BR IF OK
                                                ERROR   7               ;REPORT ERROR
                                        70$:    NOP
                                        1$:     SCOPE                   ;LOOP ?
```

```
                                        ;*****************************************************************
                                        ;*TEST 21       TEST 'PORT REQUEST' FROM PORT 'A'
                                        ;*
                                        ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
                                        ;*      DRIVE IS SEIZED BY THE OTHER PORT.
                                        ;*
                                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
                                        ;*
                                        ;*  B.  WRITE 0'S INTO RPDS1 FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
                                        ;*      SEIZED BY PORT 'B'.
                                        ;*
                                        ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
                                        ;*      SWITCHED TO PORT 'A'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
                                        ;*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
                                        ;*
                                        ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;*****************************************************************
025564                                  TST21:
025564  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
025570  001406                                  BEQ     2$              ;BR IF NOT
025572  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
025574  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
025600  012737  177777  001274          1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
025606  112737  000021  001102          2$:     MOVB    #21,$TSTNM      ;TEST NUMBER
025614  012737  025636  001106                  MOV     #TEST21,$LPADR  ;LOAD LOOP ON TEST ADDRESS
025622  012737  025636  001110                  MOV     #TEST21,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
025630  012737  007640  001176                  MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
025636  012706  001100                  TEST21: MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

025642  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
025650  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
025654  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
025662  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
025670  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
025676  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
025702  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
025710  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

                                                ;SEIZE THE DRIVE THROUGH PORT B

025716  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
025724  013737  001226  001236                  MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
025732  005060  000012                          CLR     RPDS1(R0)       ;WRITE RPDS1
025736  013737  001224  001240                  MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
025744  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
025752  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

```
 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  001154            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 ?
```

```
                        ;;*****************************************************************
                        ;*TEST 22       TEST PORT REQUEST FROM PORT 'B'
                        ;*
                        ;*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
                        ;*      DRIVE IS SEIZED BY THE OTHER PORT.
                        ;*
                        ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
                        ;*
                        ;*  B.  WRITE 0'S INTO RPDS1 FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
                        ;*      SEIZED BY PORT 'A'.
                        ;*
                        ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
                        ;*      SWITCHED TO PORT 'B'.  VERIFY THAT THE ATTENTION BIT IS SET FOR
                        ;*      PORT 'B' AND IS NOT SET FOR PORT 'A'.
                        ;*
                        ;*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                        ;*
                        ;*****************************************************************
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

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

```
5053                                    ;SET PORT REQUEST
5054
5055   027140  005060  000012                   CLR     RPDS1(R0)       ;SET PORT REQUEST FOR PORT B
5056
5057                                    ;;*****************************************************************
5058                                    ;RELEASE THROUGH PORT A.  DRIVE SHOULD SWITCH TO PORT B.
5059
5060
5061                                            ;RELEASE THE DRIVE FROM PORT A
5062
5063   027144  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5064   027152  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5065   027160  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5066
5067                                            ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
5068
5069   027166  005037  001250                   CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5070   027172  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5071   027200  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
5072   027206  060037  001122                   ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5073   027212  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5074   027220  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5075   027226  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
5076   027234  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5077   027242  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5078   027250  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
5079   027256  001404                           BEQ     66$             ;BR IF STATUS FROM PORT A ZERO
5080   027260  005737  001164                   TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
5081   027264  001401                           BEQ     66$             ;BR IF ZERO
5082   027266  104031                           ERROR   31              ;REPORT DRIVE IN NEUTRAL
5083   027270  013737  001164  001126  66$:     MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
5084   027276  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5085   027304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5086   027312  001401                           BEQ     67$             ;BR IF OK
5087   027314  104027                           ERROR   27              ;REPORT REGISTER ERROR
5088   027316  000240                   67$:    NOP
5089   027320  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5090   027326  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5091   027334  005037  001244                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5092   027340  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5093   027346  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5094   027354  060037  001122                   ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5095   027360  005037  001124                   CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5096   027364  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5097   027372  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5098   027400  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5099   027406  001414                           BEQ     68$             ;BR IF OK
5100   027410  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5101   027416  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5102   027424  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5103   027432  104016                           ERROR   16              ;TYPE MESSAGE 16
5104   027434  005137  001244                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5105   027440  000240                   68$:    NOP
5106   027442  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5107   027450  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5108   027456  005037  001244                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
```

```
5109  027462  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5110  027470  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5111  027476  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5112  027502  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
5113  027510  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5114  027516  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
5115  027524  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5116  027532  001414                          BEQ     70$             ;BR IF OK
5117  027534  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5118  027542  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
5119  027550  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5120  027556  104016                          ERROR   16              ;TYPE MESSAGE 16
5121  027560  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5122  027564  000240                  70$:    NOP
5123
5124                                          ;;*****************************************************************
5125
5126                                          ;RELEASE THE DRIVE FROM PORT B
5127
5128  027566  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5129  027574  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5130  027602  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
5131
5132                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5133
5134  027610  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5135  027614  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5136  027622  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5137  027626  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5138  027634  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5139  027642  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5140  027650  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5141  027656  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5142  027664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5143  027672  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5144  027700  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5145  027706  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5146  027714  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5147  027722  001006                          BNE     72$             ;BR IF NOT
5148  027724  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5149  027730  001045                          BNE     74$             ;BR IF NOT
5150  027732  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5151  027734  000137  030120                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
5152  027740  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5153  027746  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5154  027754  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5155  027762  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5156  027766  001414                          BEQ     73$             ;BR IF ZERO
5157  027770  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5158  027776  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5159  030004  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5160  030012  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5161  030016  001012                          BNE     74$             ;BR IF NOT
5162  030020  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5163  030026  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5164  030034  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
```

```
5165  030042  104026                        ERROR   26                      ;TYPE ERROR MESSAGE 26
5166  030044  013737  001170  001126  74$:  MOV     $TMP2,$BDDAT            ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5167  030052  013737  001224  001234        MOV     PORTA,PTNBR             ;CHANGE PORT NUMBER
5168  030060  023737  001124  001170        CMP     $GDDAT,$TMP2            ;ALL BITS OK ?
5169  030066  001401                        BEQ     75$                     ;BR IF OK FROM PORT A.
5170  030070  104007                        ERROR   7                       ;REPORT ERROR
5171  030072  013737  001172  001126  75$:  MOV     $TMP3,$BDDAT            ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5172  030100  013737  001226  001234        MOV     PORTB,PTNBR             ;CHANGE PORT NUMBER
5173  030106  023737  001124  001172        CMP     $GDDAT,$TMP3            ;SEE IF READ OK FROM PORT B.
5174  030114  001401                        BEQ     76$                     ;BR IF OK
5175  030116  104007                        ERROR   7                       ;REPORT ERROR
5176  030120  000240                  76$:  NOP
5177  030122  000004                  1$:   SCOPE                           ;LOOP ?
5178
5179
5180                                  ;;*****************************************************************
5181                                  ;*TEST 23       TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'
5182                                  ;*
5183                                  ;*VERIFY THAT READING THE CONTROL REGISTER (RPCS1) DOES NOT SET 'PORT
5184                                  ;*      REQUEST'.
5185                                  ;*
5186                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY READING RPCS1.  VERIFY THAT
5187                                  ;*      THE DRIVE HAS BEEN SEIZED.
5188                                  ;*
5189                                  ;*  B.  READ THE CONTROL REGISTER FROM PORT 'A'.  VERIFY THAT 'DVA' IS NOT
5190                                  ;*      SET.
5191                                  ;*
5192                                  ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
5193                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
5194                                  ;*
5195                                  ;;*****************************************************************
5196  030124                          TST23:
5197  030124  005737  001274                TST     KYBCTL                  ;PERFORMING ONLY SINGLE TESTS ?
5198  030130  001406                        BEQ     2$                      ;BR IF NOT
5199  030132  100002                        BPL     1$                      ;BR IF JUST ENTERED TEST
5200  030134  000137  002602                JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
5201  030140  012737  177777  001274  1$:   MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
5202  030146  112737  000023  001102  2$:   MOVB    #23,$TSTNM              ;TEST NUMBER
5203  030154  012737  030176  001106        MOV     #TEST23,$LPADR          ;LOAD LOOP ON TEST ADDRESS
5204  030162  012737  030176  001110        MOV     #TEST23,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
5205  030170  012737  007640  001176        MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS
5206  030176  012706  001100          TEST23: MOV     #STACK,SP               ;LOAD THE STACK POINTER
5207
5208                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS
5209
5210  030202  113760  001224  000010        MOVB    PORTA,RPCS2(R0)         ;SELECT PORT #A
5211  030210  005060  000012                CLR     RPDS1(R0)               ;SEIZE THE DRIVE
5212  030214  012760  000011  000000        MOV     #11,RPCS1(R0)           ;ISSUE DRIVE CLEAR
5213  030222  012760  000013  000000        MOV     #13,RPCS1(R0)           ;RELEASE THE DRIVE
5214  030230  113760  001226  000010        MOVB    PORTB,RPCS2(R0)         ;SELECT PORT #B
5215  030236  005060  000012                CLR     RPDS1(R0)               ;SEIZE THE DRIVE THROUGH PORT 'B'
5216  030242  012760  000011  000000        MOV     #11,RPCS1(R0)           ;ISSUE DRIVE CLEAR
5217  030250  012760  000013  000000        MOV     #13,RPCS1(R0)           ;RELEASE THE DRIVE
5218
5219                                  ;;*****************************************************************
5220
```

```
                                                ;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),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
030326  010037  001122                          MOV     R0,$BDADR       ;RH11 BASE ADDRESS
030332  062737  000012  001122                  ADD     #RPDS1,$BDADR   ;GENERATE BAD REGISTER ADDRESS
030340  005037  001124                          CLR     $GDDAT          ;REGISTER SHOULD BE ZERO
030344  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS THE REGISTER ZERO
030352  001403                                  BEQ     64$             ;BR IF IT IS
030354  104004                                  ERROR   4               ;REPORT THE ERROR
030356  000137  031132                          JMP     1$              ;BYPASS REST OF THE SUBTEST
030362                                  64$:
030362  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
030370  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030376  016037  000012  001126                  MOV     RPDS1(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
030404  012737  011700  001124                  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
030412  013737  001124  001166                  MOV     $GDDAT,$TMP1    ;USE GOOD DATA AS A MASK
030420  005137  001166                          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
030424  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
030432  043737  001166  001164                  BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
030440  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
030446  001401                                  BEQ     65$             ;BR IF THEY ARE
030450  104005                                  ERROR 5                 ;REPORT THE ERROR
030452  000240                          65$:    NOP
030454  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
030462  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

030470  005037  001244                          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
030474  016037  000000  001126                  MOV     RPCS1(R0),$BDDAT ;GET CONTENTS OF RPCS1
030502  012737  000000  001122                  MOV     #RPCS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
030510  060037  001122                          ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
030514  005037  001124                          CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
030520  013737  001126  001164                  MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
030526  042737  173700  001164                  BIC     #^C4077,$TMP0   ;SAVE SPECIFIED BITS
030534  023737  001124  001164                  CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
030542  001414                                  BEQ     66$             ;BR IF OK
030544  013737  001126  001174                  MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
030552  042737  004077  001174                  BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
030560  053737  001174  001124                  BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
030566  104010                                  ERROR   10              ;REPORT THE ERROR
030570  005137  001244                          COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
030574  000240                          66$:    NOP

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


                                                ;RELEASE THE DRIVE FROM PORT B
```

```
                                                MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                                MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B

                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL

                                                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
                                                MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
                                                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
                                                MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
                                                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
                                                MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
                                                MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
                                                BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
                                                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
                                                MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
                                                MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
                                                BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
                                                CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
                                                BNE     68$             ;BR IF NOT
                                                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?.
                                                BNE     70$             ;BR IF NOT
                                                ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
                                                JMP     72$             ;BYPASS THE REST OF THE CHECKS
                                        68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
                                                MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                                MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
                                                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
                                                BEQ     69$             ;BR IF ZERO
                                                MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                                MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
                                                MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
                                                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
                                                BNE     70$             ;BR IF NOT
                                        69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
                                                MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
                                                MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                                ERROR   26              ;TYPE ERROR MESSAGE 26
                                        70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
                                                MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
                                                CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
                                                BEQ     71$             ;BR IF OK FROM PORT A.
                                                ERROR   7               ;REPORT ERROR
                                        71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                                MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
                                                CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
                                                BEQ     72$             ;BR IF OK
                                                ERROR   7               ;REPORT ERROR
                                        72$:    NOP
                                        1$:     SCOPE                   ;LOOP ?

                                        ;;*****************************************************************
                                        ;*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 NEJTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;**************************************************************************
031134                                  $ST24:
031134  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
031140  001406                                  BEQ     2$              ;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
```

```
5389  031406  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
5390  031414  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
5391  031422  013737  001124  001166          MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
5392  031430  005137  001166                  COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
5393  031434  013737  001126  001164          MOV     $BDDAT,$TMP0    ;SAVE THE ACTUAL STATUS
5394  031442  043737  001166  001164          BIC     $TMP1,$TMP0     ;CLEAR UNWANTED BITS
5395  031450  023737  001124  001164          CMP     $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
5396  031456  001401                          BEQ     65$             ;BR IF THEY ARE
5397  031460  104005                          ERROR 5                 ;REPORT THE ERROR
5398  031462  000240                  65$:    NOP
5399  031464  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5400  031472  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5401
5402                                  ;;*****************************************************************
5403                                  ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST
5404
5405  031500  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5406  031504  016037  000000  001126          MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
5407  031512  012737  000000  001122          MOV     #RPCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5408  031520  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5409  031524  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5410  031530  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
5411  031536  042737  173700  001164          BIC     #^C4077,$TMP0  ;SAVE SPECIFIED BITS
5412  031544  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
5413  031552  001414                          BEQ     66$             ;BR IF OK
5414  031554  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
5415  031562  042737  004077  001174          BIC     #4077,$TMP4     ;CLEAR THE MASKED BITS
5416  031570  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
5417  031576  104010                          ERROR   10              ;REPORT THE ERROR
5418  031600  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5419  031604  000240                  66$:    NOP
5420
5421                                  ;;*****************************************************************
5422                                  ;DRIVE SHOULD RETURN TO NEUTRAL
5423
5424
5425                                  ;RELEASE THE DRIVE FROM PORT A
5426
5427  031606  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5428  031614  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5429  031622  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
5430
5431                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5432
5433  031630  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5434  031634  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5435  031642  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5436  031646  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5437  031654  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5438  031662  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5439  031670  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5440  031676  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5441  031704  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5442  031712  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5443  031720  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5444  031726  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```

```
 5445  031734  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 5446  031742  001006                          BNE     68$             ;BR IF NOT
 5447  031744  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 5448  031750  001045                          BNE     70$             ;BR IF NOT
 5449  031752  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 5450  031754  000137  032140                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
 5451  031760  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 5452  031766  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 5453  031774  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 5454  032002  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 5455  032006  001414                          BEQ     69$             ;BR IF ZERO
 5456  032010  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 5457  032016  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 5458  032024  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 5459  032032  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 5460  032036  001012                          BNE     70$             ;BR IF NOT
 5461  032040  012737  177777  001250  69$:    MOV     #-1, ELERR      ;SET 'RELEASE ERROR' INDICATOR
 5462  032046  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 5463  032054  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5464  032062  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 5465  032064  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 5466  032072  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 5467  032100  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 5468  032106  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
 5469  032110  104007                          ERROR   7               ;REPORT ERROR
 5470  032112  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 5471  032120  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 5472  032126  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 5473  032134  001401                          BEQ     72$             ;BR IF OK
 5474  032136  104007                          ERROR   7               ;REPORT ERROR
 5475  032140  000240                  72$:    NOP
 5476  032142  000004                  1$:     SCOPE                   ;LOOP ?
 5477
 5478
 5479
 5480                                  ;*****************************************************************
 5481                                  ;*TEST 25       TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
 5482                                  ;*
 5483                                  ;*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
 5484                                  ;*      IS SEIZED BY THE OTHER PORT.
 5485                                  ;*
 5486                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 5487                                  ;*
 5488                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
 5489                                  ;*
 5490                                  ;*  C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
 5491                                  ;*
 5492                                  ;*  D.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE SWITCHED
 5493                                  ;*      TO PORT 'A'.
 5494                                  ;*
 5495                                  ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
 5496                                  ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 5497                                  ;*
 5498  032144                          ;*****************************************************************
 5499  032144  005737  001274  T25:    TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 5500  032150  001406                  BEQ     2$              ;BR IF NOT
```

```
5501  032152  100002                        BPL     1$                      ;BR IF JUST ENTERED TEST
5502  032154  000137  002602                JMP     EXEC                    ;RETURN & GET NEXT TEST NUMBER
5503  032160  012737  177777  001274  1$:   MOV     #-1,KYBCTL              ;SET SINGLE TEST INDICATOR
5504  032166  112737  000025  001102  2$:   MOVB    #25,$TSTNM              ;TEST NUMBER
5505  032174  012737  032216  001106        MOV     #TEST25,$LPADR          ;LOAD LOOP ON TEST ADDRESS
5506  032202  012737  032216  001110        MOV     #TEST25,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
5507  032210  012737  007640  001176        MOV     #4000.,$TIMES           ;;DO 4000. ITERATIONS
5508  032216  012706  001100        TEST25. MOV     #STACK,SP               ;LOAD THE STACK POINTER
5509
5510                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS
5511
5512  032222  113760  001224  000010        MOVB    PORTA,RPCS2(R0)         ;SELECT PORT #A
5513  032230  005060  000012                CLR     RPDS1(R0)               ;SEIZE THE DRIVE
5514  032234  012760  000011  000000        MOV     #11,RPCS1(R0)           ;ISSUE DRIVE CLEAR
5515  032242  012760  000013  000000        MOV     #13,RPCS1(R0)           ;RELEASE THE DRIVE
5516  032250  113760  001226  000010        MOVB    PORTB,RPCS2(R0)         ;SELECT PORT #B
5517  032256  005060  000012                CLR     RPDS1(R0)               ;SEIZE THE DRIVE THROUGH PORT 'B'
5518  032262  012760  000011  000000        MOV     #11,RPCS1(R0)           ;ISSUE DRIVE CLEAR
5519  032270  012760  000013  000000        MOV     #13,RPCS1(R0)           ;RELEASE THE DRIVE
5520
5521                                ;;**************************************************************
5522
5523                                        ;SEIZE THE DRIVE THROUGH PORT B
5524
5525  032276  113760  001226  000010        MOVB    PORTB,RPCS2(R0)         ;SELECT PORT B
5526  032304  013737  001226  001236        MOV     PORTB,SEIZPT            ;STORE SEIZING PORT'S ADDRESS
5527  032312  005060  000012                CLR     RPDS1(R0)               ;WRITE RPDS1
5528  032316  113760  001224  000010        MOVB    PORTA,RPCS2(R0)         ;SELECT PORT A
5529  032324  013737  001224  001234        MOV     PORTA,PTNBR             ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5530  032332  013737  001224  001240        MOV     PORTA,OPPRT             ;'OPPOSITE' PORT ADDRESS
5531  032340  016037  000012  001126        MOV     RPDS1(R0),$BDDAT        ;SEE IF DRIVE SEIZED BY PORT B
5532  032346  010037  001122                MOV     R0,$BDADR               ;RH11 BASE ADDRESS
5533  032352  062737  000012  001122        ADD     #RPDS1,$BDADR           ;GENERATE BAD REGISTER ADDRESS
5534  032360  005037  001124                CLR     $GDDAT                  ;REGISTER SHOULD BE ZERO
5535  032364  023737  001124  001126        CMP     $GDDAT,$BDDAT           ;IS THE REGISTER ZERO
5536  032372  001403                        BEQ     64$                     ;BR IF IT IS
5537  032374  104004                        ERROR   4                       ;REPORT THE ERROR
5538  032376  000137  033350                JMP     1$                      ;BYPASS REST OF THE SUBTEST
5539  032402                        64$:
5540  032402  113760  001226  000010        MOVB    PORTB,RPCS2(R0)         ;SELECT PORT B
5541  032410  013737  001226  001234        MOV     PORTB,PTNBR             ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5542  032416  016037  000012  001126        MOV     RPDS1(R0),$BDDAT        ;SEE IF SEIZING PORT SEES CORRECT STATUS
5543  032424  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT      ;EXPECTED STATUS
5544  032432  013737  001124  001166        MOV     $GDDAT,$TMP1            ;USE GOOD DATA AS A MASK
5545  032440  005137  001166                COM     $TMP1                   ;COMPLEMENT THE EXPECTED STATUS
5546  032444  013737  001126  001164        MOV     $BDDAT,$TMP0            ;SAVE THE ACTUAL STATUS
5547  032452  043737  001166  001164        BIC     $TMP1,$TMP0             ;CLEAR UNWANTED BITS
5548  032460  023737  001124  001164        CMP     $GDDAT,$TMP0            ;ARE THE EXPECTED STATUS BITS SET ?
5549  032466  001401                        BEQ     65$                     ;BR IF THEY ARE
5550  032470  104005                        ERROR 5                         ;REPORT THE ERROR
5551  032472  000240                65$:    NOP
5552
5553                                ;;**************************************************************
5554                                ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
5555
5556  032474  113760  001224  000010        MOVB    PORTA,RPCS2(R0)         ;SELECT PORT A
```

```
5557  032502  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5558  032510  012760  000013  000000          MOV     #13,RPCS1(R0)  ;ISSUE A RELEASE COMMAND THROUGH PORT A
5559
5560                                          ;;*****************************************************************
5561                                          ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
5562
5563  032516  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
5564  032522  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
5565  032530  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5566  032536  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
5567  032542  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
5568  032546  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
5569  032554  001403                          BEQ     66$             ;BR IF OK
5570  032556  104010                          ERROR   10              ;REPORT THE ERROR
5571  032560  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
5572  032564  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
5573  032572  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
5574  032600  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
5575  032604  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
5576  032612  001404                          BEQ     67$             ;BR IF NOT
5577  032614  104011                          ERROR   11              ;REPORT THE ERROR
5578  032616  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
5579  032624  000240                  67$:    NOP
5580  032626  005737  001244                  TST     CKERR           ;WAS RPDS1 NON ZERO ?
5581  032632  001402                          BEQ     .+6             ;CONTENTS OF RPDS1 SEEN BY PORT A
5582  032634  000137  033350                  JMP     1$              ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5583
5584                                          ;;*****************************************************************
5585
5586                                          ;RELEASE THE DRIVE FROM PORT B
5587
5588  032640  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5589  032646  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5590  032654  012760  000013  000000          MOV     #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT B
5591
5592                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
5593
5594  032662  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
5595  032666  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5596  032674  012737  000012  001122          MOV     #RPDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
5597  032702  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
5598  032706  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5599  032714  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5600  032722  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
5601  032730  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
5602  032736  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5603  032744  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
5604  032752  001404                          BEQ     68$             ;BR IF STATUS FROM PORT B ZERO
5605  032754  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
5606  032760  001401                          BEQ     68$             ;BR IF ZERO
5607  032762  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
5608  032764  013737  001164  001126  68$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
5609  032772  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
5610  033000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
5611  033006  001401                          BEQ     69$             ;BR IF OK
5612  033010  104027                          ERROR   27              ;REPORT REGISTER ERROR
```

```
5613  033012  000240                    69$:    NOP
5614
5615                                    ;RELEASE THE DRIVE FROM PORT A
5616
5617  033014  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
5618  033022  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5619  033030  012760  000013  000000            MOV     #13,RPCS1(R0)  ;ISSUE RELEASE THROUGH PORT A
5620
5621                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
5622
5623  033036  005037  001250                     CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
5624  033042  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
5625  033050  060037  001122                     ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
5626  033054  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
5627  033062  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5628  033070  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
5629  033076  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
5630  033104  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5631  033112  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5632  033120  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
5633  033126  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
5634  033134  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
5635  033142  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
5636  033150  001006                            BNE     70$             ;BR IF NOT
5637  033152  005737  001164                     TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
5638  033156  001045                            BNE     72$             ;BR IF NOT
5639  033160  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
5640  033162  000137  033346                    JMP     74$             ;BYPASS THE REST OF THE CHECKS
5641  033166  013737  001170  001126  70$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
5642  033174  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5643  033202  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
5644  033210  005737  001164                     TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
5645  033214  001414                            BEQ     71$             ;BR IF ZERO
5646  033216  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
5647  033224  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
5648  033232  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
5649  033240  005737  001166                     TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
5650  033244  001012                            BNE     72$             ;BR IF NOT
5651  033246  012737  177777  001250  71$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
5652  033254  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
5653  033262  012760  000013  000000            MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
5654  033270  104026                            ERROR   26              ;TYPE ERROR MESSAGE 26
5655  033272  013737  001170  001126  72$:     MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
5656  033300  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
5657  033306  023737  001124  001170            CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
5658  033314  001401                            BEQ     73$             ;BR IF OK FROM PORT A.
5659  033316  104007                            ERROR   7               ;REPORT ERROR
5660  033320  013737  001172  001126  73$:     MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
5661  033326  013737  001226  001234            MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
5662  033334  023737  001124  001172            CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
5663  033342  001401                            BEQ     74$             ;BR IF OK
5664  033344  104007                            ERROR   7               ;REPORT ERROR
5665  033346  000240                    74$:    NOP
5666  033350  000004                    1$:     SCOPE                   ;LOOP ?
5667
5668                                    ;;*****************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;*****************************************************************
033352                                  $T26:
033352  005737  001274                        TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
033356  001406                              BEQ     2$              ;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
```

```
5725  033600  001403                          BEQ     64$                 ;BR IF IT IS
5726  033602  104004                          ERROR   4                   ;REPORT THE ERROR
5727  033604  000137  034556                  JMP     1$                  ;BYPASS REST OF THE SUBTEST
5728  033610                          64$:
5729  033610  113760  001224  000010          MOVB    PORTA,RPCS2(R0)     ;SELECT PORT A
5730  033616  013737  001224  001234          MOV     PORTA,PTNBR         ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5731  033624  016037  000012  001126          MOV     RPDS1(R0),$BDDAT    ;SEE IF SEIZING PORT SEES CORRECT STATUS
5732  033632  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;EXPECTED STATUS
5733  033640  013737  001124  001166          MOV     $GDDAT,$TMP1        ;USE GOOD DATA AS A MASK
5734  033646  005137  001166                  COM     $TMP1               ;COMPLEMENT THE EXPECTED STATUS
5735  033652  013737  001126  001164          MOV     $BDDAT,$TMP0        ;SAVE THE ACTUAL STATUS
5736  033660  043737  001166  001164          BIC     $TMP1,$TMP0         ;CLEAR UNWANTED BITS
5737  033666  023737  001124  001164          CMP     $GDDAT,$TMP0        ;ARE THE EXPECTED STATUS BITS SET ?
5738  033674  001401                          BEQ     65$                 ;BR IF THEY ARE
5739  033676  104005                          ERROR 5                     ;REPORT THE ERROR
5740  033700  000240                  65$:    NOP
5741
5742                                  ;;*****************************************************************
5743                                  ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
5744
5745  033702  113760  001226  000010          MOVB    PORTB,RPCS2(R0)     ;SELECT PORT B
5746  033710  013737  001226  001234          MOV     PORTB,PTNBR         ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5747  033716  012760  000013  000000          MOV     #13,RPCS1(R0)       ;ISSUE A RELEASE COMMAND THROUGH PORT B
5748
5749                                  ;;*****************************************************************
5750                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
5751
5752  033724  005037  001244                  CLR     CKERR               ;CLEAR THE 'CHECK ERROR' INDICATOR
5753  033730  016037  000012  001126          MOV     RPDS1(R0),$BDDAT    ;GET CONTENTS OF RPDS1
5754  033736  012737  000012  001122          MOV     #RPDS1,$BDADR       ;FORM REGISTER ADDRESS OF ERROR MESSAGE
5755  033744  060037  001122                  ADD     R0,$BDADR           ;ADD RH11 BASE ADDRESS
5756  033750  005037  001124                  CLR     $GDDAT              ;WHAT REGISTER SHOULD BE
5757  033754  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;IS THE REGISTER OK ?
5758  033762  001403                          BEQ     66$                 ;BR IF OK
5759  033764  104010                          ERROR   10                  ;REPORT THE ERROR
5760  033766  005137  001244                  COM     CKERR               ;SET THE REGISTER COMPARE ERROR INDICATOR
5761  033772  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT    ;GET THE CONTENTS OF RHCS1
5762  034000  012737  000000  001122          MOV     #RPCS1,$BDADR       ;FORM ADDRESS OF REGISTER
5763  034006  060037  001122                  ADD     R0,$BDADR           ;ADDRESS BASE
5764  034012  032737  020000  001126          BIT     #MCPE,$BDDAT        ;IS 'MCPE' SET ?
5765  034020  001404                          BEQ     67$                 ;BR IF NOT
5766  034022  104011                          ERROR   11                  ;REPORT THE ERROR
5767  034024  012760  040000  000000          MOV     #TRE,RPCS1(R0)      ;CLEAR 'MCPE'
5768  034032  000240                  67$:    NOP
5769  034034  005737  001244                  TST     CKERR               ;WAS RPDS1 NON ZERO ?
5770  034040  001402                          BEQ     .+6                 ;CONTENTS OF RPDS1 SEEN BY PORT B
5771  034042  000137  034556                  JMP     1$                  ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
5772
5773                                  ;;*****************************************************************
5774
5775                                          ;RELEASE THE DRIVE FROM PORT A
5776
5777  034046  113760  001224  000010          MOVB    PORTA,RPCS2(R0)     ;SELECT PORT A
5778  034054  013737  001224  001234          MOV     PORTA,PTNBR         ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
5779  034062  012760  000013  000000          MOV     #13,RPCS1(R0)       ;ISSUE RELEASE THROUGH PORT A
5780
```

```
                                                ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
                                                ;
005037  001250                                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
012737  111700  001124                          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
012737  000012  001122                          MOV     #RPDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
060037  001122                                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
113760  001226  000010                          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
013737  001226  001234                          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016037  000012  001164                          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
113760  001224  000010                          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
013737  001224  001234                          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016037  000012  001126                          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
001404                                          BEQ     68$             ;BR IF STATUS FROM PORT A ZERO
005737  001164                                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
001401                                          BEQ     68$             ;BR IF ZERO
104031                                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
013737  001164  001126          68$:            MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
013737  001226  001234                          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
023737  001124  001126                          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
001401                                          BEQ     69$             ;BR IF OK
104027                                          ERROR   27              ;REPORT REGISTER ERROR
000240                          69$:            NOP
                                                ;
                                                ;RELEASE THE DRIVE FROM PORT B
                                                ;
113760  001226  000010                          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
013737  001226  001234                          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012760  000013  000000                          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
                                                ;
                                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
                                                ;
005037  001250                                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
012737  000012  001122                          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
060037  001122                                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
012737  011700  001124                          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
113760  001224  000010                          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
016037  000012  001170                          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
013737  001170  001164                          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
042737  100100  001164                          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
113760  001226  000010                          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
016037  000012  001172                          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
013737  001172  001166                          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
042737  100100  001166                          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023737  001164  001166                          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
001006                                          BNE     70$             ;BR IF NOT
005737  001164                                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
001045                                          BNE     72$             ;BR IF NOT
104046                                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
000137  034554                                  JMP     74$             ;BYPASS THE REST OF THE CHECKS
013737  001170  001126          70$:            MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
013737  001226  001234                          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
113760  001226  000010                          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
005737  001164                                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
001414                                          BEQ     71$             ;BR IF ZERO
013737  001224  001234                          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013737  001172  001126                          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
```

```
 5837  034440  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 5838  034446  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 5839  034452  001012                          BNE     72$             ;BR IF NOT
 5840  034454  012737  177777  001250  71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 5841  034462  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 5842  034470  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5843  034476  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 5844  034500  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 5845  034506  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 5846  034514  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 5847  034522  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
 5848  034524  104007                          ERROR   7               ;REPORT ERROR
 5849  034526  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 5850  034534  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 5851  034542  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 5852  034550  001401                          BEQ     74$             ;BR IF OK
 5853  034552  104007                          ERROR   7               ;REPORT ERROR
 5854  034554  000240                  74$:    NOP
 5855  034556  000004                  1$:     SCOPE                   ;LOOP ?
 5856
 5857
 5858                                  ;*****************************************************************
 5859                                  ;*TEST 27       TEST SEIZE BY WRITING ATTENTION BIT
 5860                                  ;*
 5861                                  ;*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
 5862                                  ;*      (RPAS) SEIZES THE DRIVE.  VERIFY THAT REQUEST IS SET FOR THE OTHER
 5863                                  ;*      PORT.
 5864                                  ;*
 5865                                  ;*  A.  WRITE THE APPROPRIATE DRIVE BIT INTO RPAS; VERIFY THAT THE DRIVE
 5866                                  ;*      IS SEIZED.
 5867                                  ;*
 5868                                  ;*  B.  ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
 5869                                  ;*      DRIVE SWITCHES TO THE OPPOSITE PORT.  ISSUE A RELEASE THROUGH THE
 5870                                  ;*      OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.
 5871                                  ;*
 5872                                  ;*****************************************************************
 5873  034560                  TST27:
 5874  034560  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 5875  034564  001406                          BEQ     2$              ;BR IF NOT
 5876  034566  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 5877  034570  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 5878  034574  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 5879  034602  112737  000027  001102  2$:     MOVB    #27,$TSTNM      ;TEST NUMBER
 5880  034610  012737  034632  001106          MOV     #TEST27,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 5881  034616  012737  034632  001110          MOV     #TEST27,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 5882  034624  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 5883  034632  012706  001100          TEST27: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 5884
 5885                                  ;CLEAR ATTENTION BITS FOR BOTH PORTS
 5886
 5887  034636  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 5888  034644  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 5889  034650  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 5890  034656  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5891  034664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 5892  034672  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
```

```
 5893  034676  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 5894  034704  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 5895
 5896                                  ;;*****************************************************************
 5897                                  ;SELECT DRIVE OTHER THAN THAT BEING TESTED
 5898
 5899  034712  113760  001230  000010          MOVB    PORTC,RPCS2(R0) ;SELECT DRIVE NOT BEING TESTED
 5900  034720  013737  001224  001236          MOV     PORTA,SEIZPT    ;'SEIZED' PORT ADDRESS
 5901
 5902                                  ;;*****************************************************************
 5903                                  ;WRITE THE DRIVE'S ATTENTION BIT
 5904
 5905  034726  013760  001232  000016          MOV     ASR1,RPAS(R0)   ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
 5906  034734  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 5907  034742  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5908
 5909                                  ;;*****************************************************************
 5910                                  ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
 5911
 5912  034750  005760  000012                  TST     RPDS1(R0)       ;SEE THE REGISTER THROUGH PORT A ?
 5913  034754  001014                          BNE     1$              ;BR IF YES
 5914  034756  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 5915  034764  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5916  034772  005760  000012                  TST     RPDS1(R0)       ;SEE REGISTER THROUGH PORT B ?
 5917  034776  001021                          BNE     2$              ;BR IF YES
 5918  035000  104042                          ERROR   42              ;DRIVE NOT IN NEUTRAL OR SEIZED
 5919  035002  000137  036552                  JMP     4$              ;BYPASS REST OF TEST
 5920  035006                          1$:
 5921  035006  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 5922  035014  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5923  035022  005760  000012                  TST     RPDS1(R0)       ;REGISTER SHOULD BE ZERO THROUGH PORT B
 5924  035026  001002                          BNE     .+6             ;BR IF STATUS REG IS NOT ZERO
 5925  035030  000137  035702                  JMP     3$              ;STATUS REG IS ZERO
 5926  035034  104043                          ERROR   43              ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
 5927  035036  000137  036552                  JMP     4$              ;BYPASS REST OF TEST
 5928
 5929                                  ;;*****************************************************************
 5930                                  ;PORT B HAS THE DRIVE.  VERIFY THAT PORT A HAS PORT REQUEST SET
 5931
 5932  035042                          2$:
 5933  035042  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 5934  035046  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 5935  035054  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 5936  035062  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 5937  035066  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
 5938  035074  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
 5939  035102  042737  106077  001164          BIC     #^C71700,$TMP0  ;SAVE SPECIFIED BITS
 5940  035110  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
 5941  035116  001414                          BEQ     64$             ;BR IF OK
 5942  035120  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
 5943  035126  042737  071700  001174          BIC     #71700,$TMP4    ;CLEAR THE MASKED BITS
 5944  035134  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
 5945  035142  104010                          ERROR   10              ;REPORT THE ERROR
 5946  035144  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 5947  035150  000240                  64$:    NOP
 5948  035152  013737  001226  001236          MOV     PORTB,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
```

```
 5949  035160  013737  001224  001240          MOV     PORTA,OPPRT     ;SAME AS ABOVE
 5950
 5951                                          ;RELEASE THE DRIVE FROM PORT B
 5952
 5953  035166  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 5954  035174  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5955  035202  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
 5956
 5957                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
 5958
 5959  035210  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
 5960  035214  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 5961  035222  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
 5962  035230  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
 5963  035234  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 5964  035242  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5965  035250  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
 5966  035256  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 5967  035264  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5968  035272  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
 5969  035300  001404                          BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
 5970  035302  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
 5971  035306  001401                          BEQ     66$             ;BR IF ZERO
 5972  035310  104044                          ERROR   44              ;REPORT DRIVE NOT SEIZED BY PORT A
 5973  035312  013737  001164  001126  66$:   MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
 5974  035320  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
 5975  035326  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
 5976  035334  001401                          BEQ     67$             ;BR IF OK
 5977  035336  104027                          ERROR   27              ;REPORT REGISTER ERROR
 5978  035340  000240                  67$:    NOP
 5979
 5980                                          ;RELEASE THE DRIVE FROM PORT A
 5981
 5982  035342  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 5983  035350  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 5984  035356  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
 5985
 5986                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 5987
 5988  035364  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 5989  035370  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 5990  035376  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 5991  035402  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 5992  035410  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 5993  035416  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 5994  035424  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 5995  035432  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 5996  035440  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 5997  035446  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 5998  035454  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 5999  035462  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 6000  035470  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 6001  035476  001006                          BNE     68$             ;BR IF NOT
 6002  035500  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 6003  035504  001045                          BNE     70$             ;BR IF NOT
 6004  035506  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
```

```
 6005  035510  000137  035674                    JMP     72$                     ;BYPASS THE REST OF THE CHECKS
 6006  035514  013737  001170  001126  68$:      MOV     $TMP2,$BDDAT            ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 6007  035522  013737  001226  001234            MOV     PORTB,PTNBR             ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 6008  035530  113760  001226  000010            MOVB    PORTB,RPCS2(R0)         ;SELECT PORT B.
 6009  035536  005737  001164                    TST     $TMP0                   ;SEE IF STATUS EQ 0 FROM PORT A.
 6010  035542  001414                            BEQ     69$                     ;BR IF ZERO
 6011  035544  013737  001224  001234            MOV     PORTA,PTNBR             ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 6012  035552  013737  001172  001126            MOV     $TMP3,$BDDAT            ;'BAD DATA' FOR ERROR TYPE OUT
 6013  035560  113760  001224  000010            MOVB    PORTA,RPCS2(R0)         ;SELECT PORT A.
 6014  035566  005737  001166                    TST     $TMP1                   ;SEE IF STATUS EQ ZERO FROM PORT B.
 6015  035572  001012                            BNE     70$                     ;BR IF NOT
 6016  035574  012737  177777  001250  69$:      MOV     #-1,RELERR              ;SET 'RELEASE ERROR' INDICATOR
 6017  035602  012760  000011  000000            MOV     #11,RPCS1(R0)           ;CLEAR THE DRIVE
 6018  035610  012760  000013  000000            MOV     #13,RPCS1(R0)           ;RELEASE THE DRIVE
 6019  035616  104026                            ERROR   26                      ;TYPE ERROR MESSAGE 26
 6020  035620  013737  001170  001126  70$:      MOV     $TMP2,$BDDAT            ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 6021  035626  013737  001224  001234            MOV     PORTA,PTNBR             ;CHANGE PORT NUMBER
 6022  035634  023737  001124  001170            CMP     $GDDAT,$TMP2            ;ALL BITS OK ?
 6023  035642  001401                            BEQ     71$                     ;BR IF OK FROM PORT A.
 6024  035644  104007                            ERROR   7                       ;REPORT ERROR
 6025  035646  013737  001172  001126  71$:      MOV     $TMP3,$BDDAT            ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 6026  035654  013737  001226  001234            MOV     PORTB,PTNBR             ;CHANGE PORT NUMBER
 6027  035662  023737  001124  001172            CMP     $GDDAT,$TMP3            ;SEE IF READ OK FROM PORT B.
 6028  035670  001401                            BEQ     72$                     ;BR IF OK
 6029  035672  104007                            ERROR   7                       ;REPORT ERROR
 6030  035674  000240                    72$:    NOP
 6031  035676  000137  036552                    JMP     4$
 6032
 6033                                            ;;*****************************************************************
 6034                                            ;THE DRIVE IS SEIZED BY PORT A.  VERIFY THAT PORT B HAS PORT REQUEST SET
 6035
 6036  035702                            3$:
 6037  035702  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 6038  035710  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6039  035716  005037  001244                    CLR     CKERR                   ;CLEAR THE 'CHECK ERROR' INDICATOR
 6040  035722  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 6041  035730  012737  000012  001122            MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 6042  035736  060037  001122                    ADD     R0,$BDADR               ;ADD RH11 BASE ADDRESS
 6043  035742  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;WHAT REGISTER SHOULD BE
 6044  035750  013737  001126  001164            MOV     $BDDAT,$TMP0            ;MOVE REGISTER CONTENTS TO '$TMP0'
 6045  035756  042737  106077  001164            BIC     #^C71700,$TMP0          ;SAVE SPECIFIED BITS
 6046  035764  023737  001124  001164            CMP     $GDDAT,$TMP0            ;COMPARE THE BITS
 6047  035772  001414                            BEQ     73$                     ;BR IF OK
 6048  035774  013737  001126  001174            MOV     $BDDAT,$TMP4            ;COPY 'BAD DATA'
 6049  036002  042737  071700  001174            BIC     #71700,$TMP4            ;CLEAR THE MASKED BITS
 6050  036010  053737  001174  001124            BIS     $TMP4,$GDDAT            ;'OR' WITH GOOD DATA FOR TYPEOUT
 6051  036016  104010                            ERROR   10                      ;REPORT THE ERROR
 6052  036020  005137  001244                    COM     CKERR                   ;SET THE REGISTER COMPARE ERROR INDICATOR
 6053  036024  000240                    73$:    NOP
 6054  036026  013737  001224  001236            MOV     PORTA,SEIZPT            ;ADDRESS FOR ERROR MESSAGE
 6055  036034  013737  001226  001240            MOV     PORTB,OPPRT             ;SAME AS ABOVE
 6056
 6057                                            ;RELEASE THE DRIVE FROM PORT A
 6058
 6059  036042  113760  001224  000010            MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 6060  036050  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

```
6061  036056  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
6062
6063                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
6064
6065  036064  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
6066  036070  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6067  036076  012737  000012  001122          MOV     #RPDS1,$BDADR   ;REGISTER ADDRESS INCREMENT
6068  036104  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
6069  036110  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
6070  036116  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6071  036124  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
6072  036132  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
6073  036140  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6074  036146  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
6075  036154  001404                          BEQ     75$             ;BR IF STATUS FROM PORT A ZERO
6076  036156  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
6077  036162  001401                          BEQ     75$             ;BR IF ZERO
6078  036164  104044                          ERROR   44              ;REPORT DRIVE NOT SEIZED BY PORT B
6079  036166  013737  001164  001126  75$:   MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
6080  036174  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
6081  036202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
6082  036210  001401                          BEQ     76$             ;BR IF OK
6083  036212  104027                          ERROR   27              ;REPORT REGISTER ERROR
6084  036214  000240                  76$:    NOP
6085
6086                                          ;RELEASE THE DRIVE FROM PORT B
6087
6088  036216  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
6089  036224  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6090  036232  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
6091
6092                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6093
6094  036240  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6095  036244  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6096  036252  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6097  036256  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
6098  036264  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6099  036272  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6100  036300  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6101  036306  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6102  036314  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6103  036322  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6104  036330  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6105  036336  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6106  036344  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6107  036352  001006                          BNE     77$             ;BR IF NOT
6108  036354  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6109  036360  001045                          BNE     79$             ;BR IF NOT
6110  036362  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6111  036364  000137  036550                  JMP     81$             ;BYPASS THE REST OF THE CHECKS
6112  036370  013737  001170  001126  77$:   MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6113  036376  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6114  036404  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6115  036412  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6116  036416  001414                          BEQ     78$             ;BR IF ZERO
```

```
                                                ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                                ;'BAD DATA' FOR ERROR TYPE OUT
                                                ;SELECT PORT A.
                                                ;SEE IF STATUS EQ ZERO FROM PORT B.
                                                ;BR IF NOT
                                                ;SET 'RELEASE ERROR' INDICATOR
                                                ;CLEAR THE DRIVE
                                                ;RELEASE THE DRIVE
                                                ;TYPE ERROR MESSAGE 26
                                                ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
                                                ;CHANGE PORT NUMBER
                                                ;ALL BITS OK ?
                                                ;BR IF OK FROM PORT A.
                                                ;REPORT ERROR
                                                ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                                ;CHANGE PORT NUMBER
                                                ;SEE IF READ OK FROM PORT B.
                                                ;BR IF OK
                                                ;REPORT ERROR
                                                ;LOOP ?

                                                ;;*****************************************************************
                                                ;*TEST 30       TEST NO SEIZE WHEN '0' WRITTEN INTO ATTENTION BIT
                                                ;*
                                                ;*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
                                                ;*      THE DRIVE'S ATTENTION BIT.
                                                ;*
                                                ;*  A.  SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
                                                ;*      BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
                                                ;*
                                                ;*  B.  VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
                                                ;*
                                                ;;*****************************************************************
036554                                          ;T30:
036554  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
036560  001406                                  BEQ     2$              ;BR IF NOT
036562  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
036564  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
036570  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
036576  112737  000030  001102  2$:     MOVB    #30,$TSTNM      ;TEST NUMBER
036604  012737  036626  001106          MOV     #TEST30,$LPADR  ;LOAD LOOP ON TEST ADDRESS
036612  012737  036626  001110          MOV     #TEST30,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
036620  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
036626  012706  001100          TEST30: MOV     #STACK,SP       ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

036632  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
036640  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
036644  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
036652  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
036660  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
036666  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
036672  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
```

```
                                                ;;*****************************************************************
                                                ;WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST

036714  013737  001232  001164                MOV     ASR1,$TMP0      ;STORE ATTN BIT FOR PORT A
036722  005137  001164                        COM     $TMP0           ;COMPLEMENT IT
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.
```

```
                                                BEQ     68$             ;BR IF OK
                                                ERROR   7               ;REPORT ERROR
                                        68$:    NOP
                                                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 NO
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     #RPDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037536  060037  001122                        ADD     R0,$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
                                        ;
```

```
6341  037700  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6342  037704  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6343  037712  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6344  037720  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6345  037724  012737  100000  001124          MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
6346  037732  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6347  037740  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
6348  037746  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6349  037754  001414                          BEQ     70$             ;BR IF OK
6350  037756  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6351  037764  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6352  037772  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6353  040000  104041                          ERROR   41              ;TYPE MESSAGE 41
6354  040002  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6355  040006  000240                  70$:    NOP
6356
6357
6358                                          ;;*****************************************************************
6359
6360                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6361
6362  040010  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
6363  040014  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6364  040022  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6365  040026  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
6366  040034  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6367  040042  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6368  040050  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6369  040056  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6370  040064  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6371  040072  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6372  040100  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6373  040106  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6374  040114  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6375  040122  001006                          BNE     72$             ;BR IF NOT
6376  040124  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6377  040130  001045                          BNE     74$             ;BR IF NOT
6378  040132  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6379  040134  000137  040334                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6380  040140  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6381  040146  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6382  040154  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6383  040162  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6384  040166  001414                          BEQ     73$             ;BR IF ZERO
6385  040170  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6386  040176  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6387  040204  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6388  040212  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6389  040216  001012                          BNE     74$             ;BR IF NOT
6390  040220  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6391  040226  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6392  040234  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6393  040242  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6394  040244  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6395  040252  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6396  040260  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
```

T31 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE SEQ 0118

```
6397  040266  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6398  040274  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
6399  040276  104007                          ERROR   7               ;REPORT ERROR
6400  040300  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6401  040306  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6402  040314  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6403  040322  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6404  040330  001401                          BEQ     76$             ;BR IF OK
6405  040332  104007                          ERROR   7               ;REPORT ERROR
6406  040334  000240                  76$:    NOP
6407
6408                                          ;*****************************************************************
6409                                          ;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET
6410
6411  040336  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6412  040344  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6413  040352  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6414  040356  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6415  040364  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6416  040372  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6417  040376  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6418  040402  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6419  040410  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
6420  040416  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6421  040424  001414                          BEQ     77$             ;BR IF OK
6422  040426  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6423  040434  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6424  040442  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6425  040450  104052                          ERROR   52              ;TYPE MESSAGE 52
6426  040452  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6427  040456  000240                  77$:    NOP
6428
6429                                          ;CLEAR ATTENTION BIT FOR PORT A
6430
6431  040460  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT #A
6432  040466  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
6433  040472  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
6434  040500  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6435  040506  000004                  3$:     SCOPE                   ;LOOP ?
6436
6437                                          ;*****************************************************************
6438                                          ;*TEST 32       TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
6439                                          ;*
6440                                          ;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
6441                                          ;*
6442                                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
6443                                          ;*
6444                                          ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
6445                                          ;*
6446                                          ;*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
6447                                          ;*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR
6448                                          ;*      PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
6449                                          ;*
6450                                          ;*****************************************************************
6451  040510                          TST32:
6452  040510  005737  001274                  TST     @YBCTL          ;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    #32,$TSTNM          ;TEST NUMBER
                                        MOV     #TEST32,$LPADR      ;LOAD LOOP ON TEST ADDRESS
                                        MOV     #TEST32,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
                                        MOV     #4,$TIMES           ;;DO 4 ITERATIONS
                                TEST32: 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
                                ;;**************************************************************

                                        ;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

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

                                        CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
                                        MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
                                        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
                                        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

                                1$:     TST     RPDS1(R0)       ;WAIT FOR THE DRIVE TO BE RELEASED
                                        BNE     2$              ;BR IF DRIVE RELEASED
                                        TST     WATCH           ;WATCH AT ZERO ?
                                        BNE     1$              ;BR IF NOT
                                        ERROR   36              ;DRIVE NOT RELEASED WITHIN 2 SECONDS
                                2$:
                                        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
                                        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

```
6509
6510  040756  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6511  040762  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6512  040770  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6513  040776  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6514  041002  012737  040000  001124        MOV     #ERR,$GDDAT     ;WHAT REGISTER SHOULD BE
6515  041010  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6516  041016  042737  137777  001164        BIC     #^C40000,$TMP0  ;SAVE SPECIFIED BITS
6517  041024  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6518  041032  001414                        BEQ     66$             ;BR IF OK
6519  041034  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6520  041042  042737  040000  001174        BIC     #40000,$TMP4    ;CLEAR THE MASKED BITS
6521  041050  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6522  041056  104023                        ERROR   23              ;TYPE MESSAGE 23
6523  041060  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6524  041064  000240                66$:    NOP
6525
6526                                        ;;*****************************************************************
6527                                        ;THE ERROR REGISTER SHOULD CONTAIN 1'S
6528
6529  041066  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6530  041072  016037  000014  001126        MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
6531  041100  012737  000014  001122        MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6532  041106  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6533  041112  012737  177777  001124        MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
6534  041120  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
6535  041126  001403                        BEQ     68$             ;BR IF OK
6536  041130  104010                        ERROR   10              ;REPORT THE ERROR
6537  041132  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6538  041136  000240                68$:    NOP
6539
6540                                        ;;*****************************************************************
6541                                        ;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET
6542
6543  041140  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6544  041144  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6545  041152  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6546  041160  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6547  041164  012737  100000  001124        MOV     #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
6548  041172  013737  001126  001164        MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
6549  041200  042737  077777  001164        BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
6550  041206  023737  001124  001164        CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
6551  041214  001414                        BEQ     70$             ;BR IF OK
6552  041216  013737  001126  001174        MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
6553  041224  042737  100000  001174        BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
6554  041232  053737  001174  001124        BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
6555  041240  104041                        ERROR   41              ;TYPE MESSAGE 41
6556  041242  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
6557  041246  000240                70$:    NOP
6558
6559
6560                                        ;;*****************************************************************
6561
6562                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL
6563
6564  041250  005037  001250                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
```

```
6565  041254  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
6566  041262  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
6567  041266  012737  051700  001124          MOV     #51700,$GDDAT   ;COMPARSION CONSTANT
6568  041274  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6569  041302  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
6570  041310  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
6571  041316  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6572  041324  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6573  041332  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
6574  041340  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
6575  041346  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
6576  041354  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
6577  041362  001006                          BNE     72$             ;BR IF NOT
6578  041364  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
6579  041370  001045                          BNE     74$             ;BR IF NOT
6580  041372  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
6581  041374  000137  041574                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
6582  041400  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
6583  041406  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6584  041414  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
6585  041422  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
6586  041426  001414                          BEQ     73$             ;BR IF ZERO
6587  041430  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
6588  041436  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
6589  041444  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
6590  041452  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
6591  041456  001012                          BNE     74$             ;BR IF NOT
6592  041460  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
6593  041466  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
6594  041474  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
6595  041502  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
6596  041504  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
6597  041512  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
6598  041520  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
6599  041526  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
6600  041534  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
6601  041536  104007                          ERROR   7               ;REPORT ERROR
6602  041540  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
6603  041546  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
6604  041554  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
6605  041562  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
6606  041570  001401                          BEQ     76$             ;BR IF OK
6607  041572  104007                          ERROR   7               ;REPORT ERROR
6608  041574  000240                  76$:    NOP
6609
6610                                          ;;*****************************************************************
6611                                          ;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET
6612
6613  041576  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6614  041604  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6615  041612  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
6616  041616  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6617  041624  012737  000012  001122          MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6618  041632  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
6619  041636  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
6620  041642  013737  001126  001164          MOV     $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
```

T32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE SEQ 0122

```
                                                041650  042737  077777  001164          BIC     #^CATA,$TMP0    ;SAVE SPECIFIED BITS
                                                041656  023737  001124  001164          CMP     $GDDAT,$TMP0    ;COMPARE THE BITS
                                                041664  001414                          BEQ     77$             ;BR IF OK
                                                041666  013737  001126  001174          MOV     $BDDAT,$TMP4    ;COPY 'BAD DATA'
                                                041674  042737  100000  001174          BIC     #ATA,$TMP4      ;CLEAR THE MASKED BITS
                                                041702  053737  001174  001124          BIS     $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
                                                041710  104052                          ERROR   52              ;TYPE MESSAGE 52
                                                041712  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
                                                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          ^ST33:
                                                041750  005737  001274                  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
```

```
                                        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 A

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

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

                                        MOV     #-1,RPER1(R0)   ;SET ERROR BITS
                                        MOV     #13,RPCS1(R0)   ;ISSUE A RELEASE COMMAND
                                        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    PORTA,RPCS2(R0) ;SELECT PORT A
                                        MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE THROUGH PORT A
                                        JMP     2$              ;BYPASS THE REST OF THE TEST

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

                                1$:
                                        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
                                        MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                        CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
                                        MOV     RPDS1(R0),$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
```

```
 6733  042376  104024                          ERROR   24              ;TYPE MESSAGE 24
 6734  042400  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 6735  042404  000240                  68$:    NOP
 6736  042406  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 6737  042414  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6738  042422  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 6739  042426  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
 6740  042434  012737  000014  001122          MOV     #RPER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 6741  042442  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 6742  042446  012737  177777  001124          MOV     #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
 6743  042454  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 6744  042462  001403                          BEQ     70$             ;BR IF OK
 6745  042464  104010                          ERROR   10              ;REPORT THE ERROR
 6746  042466  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 6747  042472  000240                  70$:    NOP
 6748
 6749                                          ;;*****************************************************************
 6750                                          ;CLEAR THE ERRORS THROUGH PORT A
 6751
 6752  042474  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE A DRIVE CLEAR
 6753
 6754                                          ;;*****************************************************************
 6755
 6756                                          ;RELEASE THE DRIVE FROM PORT A
 6757
 6758  042502  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 6759  042510  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6760  042516  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
 6761
 6762                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 6763
 6764  042524  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 6765  042530  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 6766  042536  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 6767  042542  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 6768  042550  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 6769  042556  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 6770  042564  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 6771  042572  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 6772  042600  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 6773  042606  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 6774  042614  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 6775  042622  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 6776  042630  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 6777  042636  001006                          BNE     72$             ;BR IF NOT
 6778  042640  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 6779  042644  001045                          BNE     74$             ;BR IF NOT
 6780  042646  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 6781  042650  000137  043034                  JMP     76$             ;BYPASS THE REST OF THE CHECKS
 6782  042654  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 6783  042662  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 6784  042670  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 6785  042676  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 6786  042702  001414                          BEQ     73$             ;BR IF ZERO
 6787  042704  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 6788  042712  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
```

```
 6789  042720  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 6790  042726  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT 2.
 6791  042732  001012                          BNE     74$             ;BR IF NOT
 6792  042734  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 6793  042742  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 6794  042750  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 6795  042756  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 6796  042760  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 6797  042766  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 6798  042774  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 6799  043002  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
 6800  043004  104007                          ERROR   7               ;REPORT ERROR
 6801  043006  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 6802  043014  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 6803  043022  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 6804  043030  001401                          BEQ     76$             ;BR IF OK
 6805  043032  104007                          ERROR   7               ;REPORT ERROR
 6806  043034  000240                  76$:    NOP
 6807  043036  000004                  2$:     SCOPE                   ;LOOP ?
 6808
 6809                                  ;*****************************************************************
 6810                                  ;*TEST 34       TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
 6811                                  ;*
 6812                                  ;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
 6813                                  ;*      BITS ARE SET IN THE DRIVE.
 6814                                  ;*
 6815                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 6816                                  ;*
 6817                                  ;*  B.  WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
 6818                                  ;*
 6819                                  ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE 'GO'
 6820                                  ;*      BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
 6821                                  ;*      THAT RPER1 HAS NOT BEEN CLEARED.
 6822                                  ;*
 6823                                  ;*  D.  CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
 6824                                  ;*
 6825                                  ;*  E.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
 6826                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 6827                                  ;*
 6828                                  ;*****************************************************************
 6829  043040                  TS^34:
 6830  043040  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 6831  043044  001406                          BEQ     2$              ;BR IF NOT
 6832  043046  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 6833  043050  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 6834  043054  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 6835  043062  112737  000034  001102  2$:     MOVB    #34,$TSTNM      ;TEST NUMBER
 6836  043070  012737  043112  001106          MOV     #TEST34,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 6837  043076  012737  043112  001110          MOV     #TEST34,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 6838  043104  012737  007640  001176          MOV     #4000.,$TIMES   ;;DO 4000. ITERATIONS
 6839  043112  012706  001100          TEST34: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 6840
 6841                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 6842
 6843  043116  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 6844  043124  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
```

```
6845  043130  012760  000011  000000          MOV     #11,RPCS1(R0)    ;ISSUE DRIVE CLEAR
6846  043136  012760  000013  000000          MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
6847  043144  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT #B
6848  043152  005060  000012                  CLR     RPDS1(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
6849  043156  012760  000011  000000          MOV     #11,RPCS1(R0)    ;ISSUE DRIVE CLEAR
6850  043164  012760  000013  000000          MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
6851                                  ;;**************************************************************
6852
6853                                  ;SEIZE THE DRIVE THROUGH PORT B
6854
6855  043172  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6856  043200  013737  001226  001236          MOV     PORTB,SEIZPT     ;STORE SEIZING PORT'S ADDRESS
6857  043206  005060  000012                  CLR     RPDS1(R0)        ;WRITE RPDS1
6858  043212  013737  001224  001240          MOV     PORTA,OPPRT      ;'OPPOSITE' PORT ADDRESS
6859
6860                                  ;;**************************************************************
6861                                  ;FORCE AN ERROR
6862
6863  043220  012760  177777  000014          MOV     #-1,RPER1(R0)    ;SET ERROR BITS
6864  043226  012760  000013  000000          MOV     #13,RPCS1(R0)    ;ISSUE A RELEASE COMMAND
6865  043234  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
6866  043240  016037  000000  001126          MOV     RPCS1(R0),$BDDAT  ;GET CONTENTS OF RPCS1
6867  043246  012737  000000  001122          MOV     #RPCS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6868  043254  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
6869  043260  012737  004012  001124          MOV     #4012,$GDDAT     ;WHAT REGISTER SHOULD BE
6870  043266  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
6871  043274  042737  173765  001164          BIC     #^C4012,$TMP0    ;SAVE SPECIFIED BITS
6872  043302  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
6873  043310  001414                          BEQ     66$              ;BR IF OK
6874  043312  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
6875  043320  042737  004012  001174          BIC     #4012,$TMP4      ;CLEAR THE MASKED BITS
6876  043326  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
6877  043334  104025                          ERROR   25               ;TYPE MESSAGE 25
6878  043336  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
6879  043342  000240                  66$:    NOP
6880  043344  005737  001244                  TST     CKERR            ;DID 'GO' BIT RESET ?
6881  043350  001002                          BNE     .+6              ;BR IF NOT
6882  043352  000137  043412                  JMP     1$               ;'GO' BIT RESET
6883  043356  012760  000040  000010          MOV     #CLR,RPCS2(R0)   ;INIT THE RH11
6884  043364  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
6885  043372  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6886  043400  012760  000013  000000          MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE THROUGH PORT B
6887  043406  000137  044126                  JMP     2$               ;BYPASS THE REST OF THE TEST
6888
6889                                  ;;**************************************************************
6890                                  ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
6891
6892  043412                          1$:
6893  043412  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
6894  043420  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
6895  043426  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
6896  043432  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
6897  043440  012737  000012  001122          MOV     #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
6898  043446  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
6899  043452  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
6900  043456  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;IS THE REGISTER OK ?
```

```
 6901  043464  001403                          BEQ     68$                ;BR IF OK
 6902  043466  104024                          ERROR   24                 ;TYPE MESSAGE 24
 6903  043470  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
 6904  043474  000240                  68$:    NOP
 6905  043476  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B
 6906  043504  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6907  043512  005037  001244                  CLR     CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
 6908  043516  016037  000014  001126          MOV     RPER1(R0),$BDDAT   ;GET CONTENTS OF RPER1
 6909  043524  012737  000014  001122          MOV     #RPER1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 6910  043532  060037  001122                  ADD     R0,$BDADR          ;ADD RH11 BASE ADDRESS
 6911  043536  012737  177777  001124          MOV     #177777,$GDDAT     ;WHAT REGISTER SHOULD BE
 6912  043544  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;IS THE REGISTER OK ?
 6913  043552  001403                          BEQ     70$                ;BR IF OK
 6914  043554  104010                          ERROR   10                 ;REPORT THE ERROR
 6915  043556  005137  001244                  COM     CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
 6916  043562  000240                  70$:    NOP
 6917
 6918                                          ;;*****************************************************************
 6919                                          ;CLEAR THE ERRORS THROUGH PORT B
 6920
 6921  043564  012760  000011  000000          MOV     #11,RPCS1(R0)      ;ISSUE A DRIVE CLEAR
 6922
 6923                                          ;;*****************************************************************
 6924
 6925                                          ;RELEASE THE DRIVE FROM PORT B
 6926
 6927  043572  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B
 6928  043600  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 6929  043606  012760  000013  000000          MOV     #13,RPCS1(R0)      ;ISSUE RELEASE THROUGH PORT B
 6930
 6931                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 6932
 6933  043614  005037  001250                  CLR     RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 6934  043620  012737  000012  001122          MOV     #RPDS1,$BDADR      ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 6935  043626  060037  001122                  ADD     R0,$BDADR          ;ADD THE I/O BASE ADDRESS
 6936  043632  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 6937  043640  113760  001224  000010          MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A.
 6938  043646  016037  000012  001170          MOV     RPDS1(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 6939  043654  013737  001170  001164          MOV     $TMP2,$TMP0        ;COPY IT INTO '$TMP0'
 6940  043662  042737  100100  001164          BIC     #ATA!VV,$TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 6941  043670  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B.
 6942  043676  016037  000012  001172          MOV     RPDS1(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 6943  043704  013737  001172  001166          MOV     $TMP3,$TMP1        ;COPY IT INTO '$TMP1'
 6944  043712  042737  100100  001166          BIC     #ATA!VV,$TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 6945  043720  023737  001164  001166          CMP     $TMP0,$TMP1        ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 6946  043726  001006                          BNE     72$                ;BR IF NOT
 6947  043730  005737  001164                  TST     $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 6948  043734  001045                          BNE     74$                ;BR IF NOT
 6949  043736  104046                          ERROR   46                 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 6950  043740  000137  044124                  JMP     76$                ;BYPASS THE REST OF THE CHECKS
 6951  043744  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT       ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 6952  043752  013737  001226  001234          MOV     PORTB,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 6953  043760  113760  001226  000010          MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B.
 6954  043766  005737  001164                  TST     $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
 6955  043772  001414                          BEQ     73$                ;BR IF ZERO
 6956  043774  013737  001224  001234          MOV     PORTA,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
 6957  044002  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 6958  044010  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 6959  044016  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 6960  044022  001012                          BNE     74$             ;BR IF NOT
 6961  044024  012737  177777  001250  73$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 6962  044032  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 6963  044040  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 6964  044046  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 6965  044050  013737  001170  001126  74$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 6966  044056  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 6967  044064  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 6968  044072  001401                          BEQ     75$             ;BR IF OK FROM PORT A.
 6969  044074  104007                          ERROR   7               ;REPORT ERROR
 6970  044076  013737  001172  001126  75$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 6971  044104  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 6972  044112  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 6973  044120  001401                          BEQ     76$             ;BR IF OK
 6974  044122  104007                          ERROR   7               ;REPORT ERROR
 6975  044124  000240                  76$:    NOP
 6976  044126  000004                  2$:     SCOPE                   ;LOOP ?
 6977
 6978
 6979                                  ;*****************************************************************
 6980                                  ;*TEST 35       TEST TIMEOUT RETRIGGER THROUGH PORT 'A'
 6981                                  ;*
 6982                                  ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
 6983                                  ;*
 6984                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 6985                                  ;*
 6986                                  ;*  B.  WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.
 6987                                  ;*
 6988                                  ;*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
 6989                                  ;*      TIME.  (THE MEASUREMENT IS MADE FROM STEP 'B'.)
 6990                                  ;*
 6991                                  ;*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
 6992                                  ;*      BIT IS SET.
 6993                                  ;*
 6994                                  ;*****************************************************************
 6995  044130                  TST35:
 6996  044130  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 6997  044134  001406                          BEQ     2$              ;BR IF NOT
 6998  044136  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 6999  044140  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 7000  044144  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 7001  044152  112737  000035  001102  2$:     MOVB    #35,$TSTNM      ;TEST NUMBER
 7002  044160  012737  044202  001106          MOV     #TEST35,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 7003  044166  012737  044202  001110          MOV     #TEST35,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 7004  044174  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 7005  044202  012706  001100          TEST35: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 7006
 7007                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 7008
 7009  044206  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 7010  044214  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 7011  044220  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 7012  044226  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
```

```
7013  044234  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
7014  044242  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
7015  044246  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
7016  044254  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7017
7018                                  ;;*****************************************************************
7019
7020                                  ;SEIZE THE DRIVE THROUGH PORT A
7021
7022  044262  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
7023  044270  013737  001224  001236          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
7024  044276  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
7025  044302  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
7026
7027                                  ;;*****************************************************************
7028                                  ;WAIT 500 MS
7029
7030
7031                                  ;;*****************************************************************
7032                                  ;START THE TIMER
7033
7034  044310  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7035  044314  012737  000764  001254          MOV     #500.,WATCH     ;SET WATCH TO 500 MS
7036  044322  005737  001254          1$:     TST     WATCH           ;WATCH EQUAL TO ZERO
7037  044326  001375                  BNE     1$              ;BR IF NOT
7038
7039                                  ;;*****************************************************************
7040                                  ;START THE TIMER
7041
7042  044330  005037  001252                  CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
7043  044334  012737  003720  001254          MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
7044
7045                                  ;;*****************************************************************
7046                                  ;RETRIGGER THE TIMEOUT ONE-SHOT
7047
7048  044342  005760  000012                  TST     RPDS1(R0)       ;RETRIGGER THE ONE-SHOT
7049  044346  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
7050  044354  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7051  044362  005760  000012          2$:     TST     RPDS1(R0)       ;WAIT FOR TIMEOUT
7052  044366  001004                  BNE     3$              ;BR IF TIMEOUT OCCURRED
7053  044370  005737  001254                  TST     WATCH           ;WATCH EQUAL TO ZERO ?
7054  044374  001372                  BNE     2$              ;BR IF NOT
7055  044376  104036                  ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7056  044400  013737  001252  001272  3$:     MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7057
7058                                  ;;*****************************************************************
7059
7060                                  ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7061
7062  044406  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7063  044412  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7064  044420  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7065  044424  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7066  044432  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7067  044440  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7068  044446  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
```

```
 7069  044454  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 7070  044462  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 7071  044470  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 7072  044476  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 7073  044504  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 7074  044512  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 7075  044520  001006                          BNE     66$             ;BR IF NOT
 7076  044522  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 7077  044526  001045                          BNE     68$             ;BR IF NOT
 7078  044530  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 7079  044532  000137  044716                  JMP     70$             ;BYPASS THE REST OF THE CHECKS
 7080  044536  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 7081  044544  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 7082  044552  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 7083  044560  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 7084  044564  001414                          BEQ     67$             ;BR IF ZERO
 7085  044566  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 7086  044574  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 7087  044602  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 7088  044610  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 7089  044614  001012                          BNE     68$             ;BR IF NOT
 7090  044616  012737  177777  001250  67$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 7091  044624  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 7092  044632  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 7093  044640  104022                          ERROR   22              ;TYPE ERROR MESSAGE 22
 7094  044642  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 7095  044650  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 7096  044656  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 7097  044664  001401                          BEQ     69$             ;BR IF OK FROM PORT A.
 7098  044666  104007                          ERROR   7               ;REPORT ERROR
 7099  044670  013737  001172  001126  69$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 7100  044676  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 7101  044704  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 7102  044712  001401                          BEQ     70$             ;BR IF OK
 7103  044714  104007                          ERROR   7               ;REPORT ERROR
 7104  044716  000240                  70$:    NOP
 7105
 7106                                  ;;*****************************************************************
 7107                                  ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
 7108
 7109  044720  023737  001272  001260          CMP     TIMES,TIMEAP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
 7110  044726  003004                          BGT     4$              ;BR IF GREATER
 7111  044730  023737  001272  001262          CMP     TIMES,TIMEAM    ;MEASURED TIME LESS THAN -25% TOLERANCE
 7112  044736  002001                          BGE     .+4             ;BR IF NOT
 7113  044740  104025                  4$:     ERROR   25              ;REPORT THE ERROR
 7114  044742  000004                          SCOPE                   ;LOOP ?
 7115
 7116                                  ;;*****************************************************************
 7117                                  ;*TEST 36       TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
 7118                                  ;*
 7119                                  ;*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
 7120                                  ;*
 7121                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 7122                                  ;*
 7123                                  ;*  B.  WAIT 500 MS AND WRITE 0'B INTO RPDS1 THROUGH PORT 'A'.
 7124                                  ;*
```

```
                                        ;*  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.
                                        ;*
                                        ;;*****************************************************************
044744                                  ↑ST36:
044744  005737  001274                          TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
044750  001406                                  BEQ     2$              ;BR IF NOT
044752  100002                                  BPL     1$              ;BR IF JUST ENTERED TEST
044754  000137  002602                          JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
044760  012737  177777  001274      1$:         MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
044766  112737  000036  001102      2$:         MOVB    #36,$TSTNM      ;TEST NUMBER
044774  012737  045016  001106                  MOV     #TEST36,$LPADR  ;LOAD LOOP ON TEST ADDRESS
045002  012737  045016  001110                  MOV     #TEST36,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
045010  012737  000004  001176                  MOV     #4,$TIMES       ;;DO 4 ITERATIONS
045016  012706  001100              TEST36:     MOV     #↑STACK,SP      ;LOAD THE STACK POINTER

                                        ;CLEAR ATTENTION BITS FOR BOTH PORTS

045022  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
045030  005060  000012                      CLR     RPDS1(R0)       ;SEIZE THE DRIVE
045034  012760  000011  000000              MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
045042  012760  000013  000000              MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
045050  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
045056  005060  000012                      CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
045062  012760  000011  000000              MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
045070  012760  000013  000000              MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

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

                                        ;SEIZE THE DRIVE THROUGH PORT B

045076  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
045104  013737  001226  001236              MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
045112  005060  000012                      CLR     RPDS1(R0)       ;WRITE RPDS1
045116  013737  001224  001240              MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

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


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

045124  005037  001252                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
045130  012737  000764  001254                  MOV     #500.,WATCH     ;SET WATCH TO 500 MS
045136  005737  001254              1$:         TST     WATCH           ;WATCH EQUAL TO ZERO
045142  001375                                  BNE     1$              ;BR IF NOT

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

045144  005037  001252                          CLR     TIME            ;CLEAR THE ELAPSED TIME COUNTER
045150  012737  003720  001254                  MOV     #2000.,WATCH    ;SET WATCH TO 2000 MS
```

```
7181
7182                                    ;;*****************************************************************
7183                                    ;RETRIGGER THE TIMEOUT ONE-SHOT
7184
7185  045156  005760  000012                      TST     RPDS1(R0)       ;RETRIGGER THE ONE-SHOT
7186  045162  113760  001224  000010              MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7187  045170  013737  001224  001234              MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7188  045176  005760  000012              2$:     TST     RPDS1(R0)       ;WAIT FOR TIMEOUT
7189  045202  001004                      BNE     3$              ;BR IF TIMEOUT OCCURRED
7190  045204  005737  001254              TST     WATCH           ;WATCH EQUAL TO ZERO ?
7191  045210  001372                      BNE     2$              ;BR IF NOT
7192  045212  104036                      ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
7193  045214  013737  001252  001272      3$:     MOV     TIME,TIMES      ;SAVE THE ELAPSED TIME VALUE
7194
7195                                    ;;*****************************************************************
7196
7197                                    ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7198
7199  045222  005037  001250                      CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7200  045226  012737  000012  001122              MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7201  045234  060037  001122                      ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7202  045240  012737  011700  001124              MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7203  045246  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7204  045254  016037  000012  001170              MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7205  045262  013737  001170  001164              MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7206  045270  042737  100100  001164              BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7207  045276  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7208  045304  016037  000012  001172              MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7209  045312  013737  001172  001166              MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7210  045320  042737  100100  001166              BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7211  045326  023737  001164  001166              CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7212  045334  001006                      BNE     66$             ;BR IF NOT
7213  045336  005737  001164              TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7214  045342  001045                      BNE     68$             ;BR IF NOT
7215  045344  104046                      ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7216  045346  000137  045532              JMP     70$             ;BYPASS THE REST OF THE CHECKS
7217  045352  013737  001170  001126  66$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7218  045360  013737  001226  001234              MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7219  045366  113760  001226  000010              MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7220  045374  005737  001164              TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7221  045400  001414                      BEQ     67$             ;BR IF ZERO
7222  045402  013737  001224  001234              MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7223  045410  013737  001172  001126              MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7224  045416  113760  001224  000010              MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7225  045424  005737  001166              TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7226  045430  001012                      BNE     68$             ;BR IF NOT
7227  045432  012737  177777  001250  67$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7228  045440  012760  000011  000000              MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7229  045446  012760  000013  000000              MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7230  045454  104022                      ERROR   22              ;TYPE ERROR MESSAGE 22
7231  045456  013737  001170  001126  68$:     MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7232  045464  013737  001224  001234              MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7233  045472  023737  001124  001170              CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7234  045500  001401                      BEQ     69$             ;BR IF OK FROM PORT A.
7235  045502  104007                      ERROR   7               ;REPORT ERROR
7236  045504  013737  001172  001126  69$:     MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
```

```
 7237  045512  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 7238  045520  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 7239  045526  001401                          BEQ     70$             ;BR IF OK
 7240  045530  104007                          ERROR   7               ;REPORT ERROR
 7241  045532  000240                  70$:    NOP
 7242
 7243                                          ;*****************************************************************
 7244                                          ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
 7245
 7246  045534  023737  001272  001266          CMP     TIMES,TIMEBP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
 7247  045542  003004                          BGT     4$              ;BR IF GREATER
 7248  045544  023737  001272  001270          CMP     TIMES,TIMEBM    ;MEASURED TIME LESS THAN -25% TOLERANCE
 7249  045552  002001                          BGE     .+4             ;BR IF NOT
 7250  045554  104025                  4$:     ERROR   25              ;REPORT THE ERROR
 7251  045556  000004                          SCOPE                   ;LOOP ?
 7252
 7253
 7254
 7255                                          ;*****************************************************************
 7256                                          ;*TEST 37       TEST PORT 'A' ATTENTION AFTER A COMMAND
 7257                                          ;*
 7258                                          ;*
 7259                                          ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
 7260                                          ;*      COMMAND.
 7261                                          ;*
 7262                                          ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
 7263                                          ;*
 7264                                          ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
 7265                                          ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
 7266                                          ;*      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
 7267                                          ;*
 7268                                          ;*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
 7269                                          ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 7270                                          ;*
 7271                                          ;*****************************************************************
 7272  045560                          TST37:
 7273  045560  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 7274  045564  001406                          BEQ     2$              ;BR IF NOT
 7275  045566  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 7276  045570  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 7277  045574  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 7278  045602  112737  000037  001102  2$:     MOVB    #37,$TSTNM      ;TEST NUMBER
 7279  045610  012737  045632  001106          MOV     #TEST37,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 7280  045616  012737  045632  001110          MOV     #TEST37,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 7281  045624  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 7282  045632  012706  001100          TEST37: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 7283
 7284                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 7285
 7286  045636  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 7287  045644  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 7288  045650  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 7289  045656  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 7290  045664  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 7291  045672  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 7292  045676  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
```

```
 7293  045704  012760  000013  000000          MOV     #13,RPCS1(R0)    ;RELEASE THE DRIVE
 7294  045712  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 7295  045720  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 7296  045726  013737  001224  001236          MOV     PORTA,SEIZPT     ;'SEIZED' PORT ADDRESS
 7297
 7298                                  ;;*****************************************************************
 7299                                  ;DO A RECALIBRATE THROUGH PORT A
 7300
 7301  045734  012760  000007  000000          MOV     #7,RPCS1(R0)     ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
 7302
 7303                                  ;;*****************************************************************
 7304                                  ;WAIT FOR DRIVE TO FINISH
 7305
 7306  045742  032760  000200  000012          BIT     #DRY,RPDS1(R0)   ;WAIT FOR DRIVE TO FINISH
 7307  045750  001774                          BEQ     .-6              ;BR IF NOT FINISHED
 7308
 7309                                  ;;*****************************************************************
 7310                                  ;CONFIRM THAT ATTENTION IS SET FOR PORT A
 7311
 7312  045752  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
 7313  045756  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 7314  045764  012737  000012  001122          MOV     #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7315  045772  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
 7316  045776  012737  100000  001124          MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
 7317  046004  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
 7318  046012  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
 7319  046020  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
 7320  046026  001414                          BEQ     64$              ;BR IF OK
 7321  046030  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
 7322  046036  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
 7323  046044  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
 7324  046052  104032                          ERROR   32               ;TYPE MESSAGE 32
 7325  046054  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
 7326  046060  000240                  64$:    NOP
 7327  046062  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
 7328  046070  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 7329
 7330                                  ;;*****************************************************************
 7331                                  ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B
 7332
 7333  046076  005037  001244                  CLR     CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
 7334  046102  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 7335  046110  012737  000012  001122          MOV     #RPDS1,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7336  046116  060037  001122                  ADD     R0,$BDADR        ;ADD RH11 BASE ADDRESS
 7337  046122  005037  001124                  CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
 7338  046126  013737  001126  001164          MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
 7339  046134  042737  077777  001164          BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
 7340  046142  023737  001124  001164          CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
 7341  046150  001414                          BEQ     66$              ;BR IF OK
 7342  046152  013737  001126  001174          MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
 7343  046160  042737  100000  001174          BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
 7344  046166  053737  001174  001124          BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
 7345  046174  104032                          ERROR   32               ;TYPE MESSAGE 32
 7346  046176  005137  001244                  COM     CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
 7347  046202  000240                  66$:    NOP
 7348
```

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

                                        ;RELEASE THE DRIVE FROM PORT A

046204  113760  001224  000010        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
046212  013737  001224  001234        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046220  012760  000013  000000        MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL

046226  005037  001250                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
046232  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
046240  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
046244  012737  011700  001124        MOV     #MOL!PGM'DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
046252  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
046260  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
046266  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
046274  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046302  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
046310  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046316  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
046324  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046332  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046340  001006                        BNE     68$             ;BR IF NOT
046342  005737  001164                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046346  001045                        BNE     70$             ;BR IF NOT
046350  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
046352  000137  046536                JMP     72$             ;BYPASS THE REST OF THE CHECKS
046356  013737  001170  001126  68$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046364  013737  001226  001234        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046372  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
046400  005737  001164                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
046404  001414                        BEQ     69$             ;BR IF ZERO
046406  013737  001224  001234        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046414  013737  001172  001126        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
046422  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
046430  005737  001166                TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
046434  001012                        BNE     70$             ;BR IF NOT
046436  012737  177777  001250  69$:  MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
046444  012760  000011  000000        MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
046452  012760  000013  000000        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
046460  104026                        ERROR   26              ;TYPE ERROR MESSAGE 26
046462  013737  001170  001126  70$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
046470  013737  001224  001234        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
046476  023737  001124  001170        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
046504  001401                        BEQ     71$             ;BR IF OK FROM PORT A.
046506  104007                        ERROR   7               ;REPORT ERROR
046510  013737  001172  001126  71$:  MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
046516  013737  001226  001234        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
046524  023737  001124  001172        CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
046532  001401                        BEQ     72$             ;BR IF OK
046534  104007                        ERROR   7               ;REPORT ERROR
046536  000240                72$:    NOP
046540  000004                        SCOPE                   ;LOOP ?

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

```
                                        ;*TEST 40       TEST PORT 'B' ATTENTION AFTER A COMMAND
                                        ;*
                                        ;*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
                                        ;*      COMMAND.
                                        ;*
                                        ;*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
                                        ;*
                                        ;*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
                                        ;*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND
                                        ;*      THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
                                        ;*
                                        ;*  C.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
                                        ;*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;;*****************************************************************
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
```

```
                                        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
                                        MOV     #RPDS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
                                        ADD     R0,$BDADR         ;ADD RH11 BASE ADDRESS
                                        MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
                                        MOV     $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
                                        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
                                        CMP     $GDDAT,$TMP0      ;COMPARE THE BITS
                                        BEQ     64$               ;BR IF OK
                                        MOV     $BDDAT,$TMP4      ;COPY 'BAD DATA'
                                        BIC     #ATA,$TMP4        ;CLEAR THE MASKED BITS
                                        BIS     $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
                                        ERROR   32                ;TYPE MESSAGE 32
                                        COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
                                64$:    NOP
                                        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
                                        MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

                                ;;*****************************************************************
                                ;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

                                        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
                                        MOV     $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
                                        BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
                                        CMP     $GDDAT,$TMP0      ;COMPARE THE BITS
                                        BEQ     66$               ;BR IF OK
                                        MOV     $BDDAT,$TMP4      ;COPY 'BAD DATA'
                                        BIC     #ATA,$TMP4        ;CLEAR THE MASKED BITS
                                        BIS     $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
                                        ERROR   32                ;TYPE MESSAGE 32
                                        COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
                                66$:    NOP

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

                                        ;RELEASE THE DRIVE FROM PORT B

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

                                        ;VERIFY THAT THE DRIVE IS IN NEUTRAL

                                        CLR     RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
                                        MOV     #RPDS1,$BDADR     ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
                                        ADD     R0,$BDADR         ;ADD THE I/O BASE ADDRESS
                                        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
                                        MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A.
                                        MOV     RPDS1(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
                                        MOV     $TMP2,$TMP0       ;COPY IT INTO '$TMP0'
                                        BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
                                        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B.
                                        MOV     RPDS1(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
```

```
7517  047300  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7518  047306  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7519  047314  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7520  047322  001006                          BNE     68$             ;BR IF NOT
7521  047324  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7522  047330  001045                          BNE     70$             ;BR IF NOT
7523  047332  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7524  047334  000137  047520                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
7525  047340  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7526  047346  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7527  047354  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7528  047362  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
7529  047366  001414                          BEQ     69$             ;BR IF ZERO
7530  047370  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
7531  047376  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
7532  047404  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7533  047412  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
7534  047416  001012                          BNE     70$             ;BR IF NOT
7535  047420  012737  177777  001250  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
7536  047426  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
7537  047434  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
7538  047442  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
7539  047444  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
7540  047452  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
7541  047460  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
7542  047466  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
7543  047470  104007                          ERROR   7               ;REPORT ERROR
7544  047472  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
7545  047500  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
7546  047506  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
7547  047514  001401                          BEQ     72$             ;BR IF OK
7548  047516  104007                          ERROR   7               ;REPORT ERROR
7549  047520  000240                  72$:    NOP
7550  047522  000004                          SCOPE                   ;LOOP ?
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
```

```
;*****************************************************************
;*TEST 41       TEST PORT INTERACTION FROM PORT 'A'
;*
;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
;*
;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
;*
;*  B.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
;*
;*  C.  READ RPER1, RPER2, & RPER3 THROUGH PORT 'B'.  VERIFY THAT PORT
;*      'B' SEES 0'S FROM EACH OF THESE REGISTERS.
;*
;*  D.  CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'A'.
;*
;*  E.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.  VERIFY THAT
;*      PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
;*
;*  F.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE HAS
;*      SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS
;*      SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
```

```
                                        ;*
                                        ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
                                        ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
                                        ;*
                                        ;**************************************************************
047524                                  TST41:
047524  005737  001274                          TST     KYBCTL              ;PERFORMING ONLY SINGLE TESTS ?
047530  001406                                  BEQ     2$                  ;BR IF NOT
047532  100002                                  BPL     1$                  ;BR IF JUST ENTERED TEST
047534  000137  002602                          JMP     EXEC                ;RETURN & GET NEXT TEST NUMBER
047540  012737  177777  001274          1$:     MOV     #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
047546  112737  000041  001102          2$:     MOVB    #41,$TSTNM          ;TEST NUMBER
047554  012737  047576  001106                  MOV     #TEST41,$LPADR      ;LOAD LOOP ON TEST ADDRESS
047562  012737  047576  001110                  MOV     #TEST41,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
047570  012737  007640  001176                  MOV     #4000.,$TIMES       ;;DO 4000. ITERATIONS
047576  012706  001100                  TEST41: MOV     #STACK,SP           ;LOAD THE STACK POINTER

                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

047602  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
047610  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE
047614  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
047622  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
047630  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
047636  005060  000012                          CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
047642  012760  000011  000000                  MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
047650  012760  000013  000000                  MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE

                                                ;SEIZE THE DRIVE THROUGH PORT A

047656  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
047664  013737  001224  001236                  MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
047672  005060  000012                          CLR     RPDS1(R0)       ;WRITE RPDS1
047676  013737  001226  001240                  MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
047704  012760  177777  000014                  MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT A
047712  012760  177777  000040                  MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT A
047720  012760  177777  000042                  MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT A
047726  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
047734  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047742  004737  050600                          JSR     PC,TST41B       ;CHECK THE REGISTERS THROUGH PORT B
047746  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
047754  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047762  005060  000042                          CLR     RPER3(R0)       ;CLEAR RPER3 ON PORT A
047766  005060  000040                          CLR     RPER2(R0)       ;CLEAR RPER2 ON PORT A
047772  005060  000014                          CLR     RPER1(R0)       ;CLEAR RPER1 ON PORT A
047776  013760  001232  000016                  MOV     ASR1,RPAS(R0)   ;CLEAR THE ATTENTION BIT FOR PORT A
050004  113760  001226  000010                  MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
050012  013737  001226  001234                  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050020  012760  177777  000014                  MOV     #-1,RPER1(R0)   ;LOAD 1'S INTO RPER1 THROUGH PORT B
050026  012760  177777  000040                  MOV     #-1,RPER2(R0)   ;LOAD 1'S INTO RPER2 THROUGH PORT B
050034  012760  177777  000042                  MOV     #-1,RPER3(R0)   ;LOAD 1'S INTO RPER3 THROUGH PORT B
050042  113760  001224  000010                  MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
050050  013737  001224  001234                  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050056  004737  050600                          JSR     PC,TST41B       ;CHECK THE REGISTERS THROUGH PORT A

                                                ;RELEASE THE DRIVE FROM PORT A
```

```
7629
7630  050062  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7631  050070  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7632  050076  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
7633
7634                                          ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
7635
7636  050104  005037  001250                  CLR     RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
7637  050110  012737  111700  001124          MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7638  050116  012737  000012  001122          MOV     #RPDS1,$BDADR  ;REGISTER ADDRESS INCREMENT
7639  050124  060037  001122                  ADD     R0,$BDADR       ;REGISTER BASE ADDRESS FOR TYPEOUT
7640  050130  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7641  050136  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7642  050144  016037  000012  001164          MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
7643  050152  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
7644  050160  013737  001224  001234          MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7645  050166  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT A
7646  050174  001404                          BEQ     66$             ;BR IF STATUS FROM PORT A ZERO
7647  050176  005737  001164                  TST     $TMP0           ;IS STATUS FROM PORT B ZERO ?
7648  050202  001401                          BEQ     66$             ;BR IF ZERO
7649  050204  104031                          ERROR   31              ;REPORT DRIVE IN NEUTRAL
7650  050206  013737  001164  001126  66$:    MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT B
7651  050214  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
7652  050222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
7653  050230  001401                          BEQ     67$             ;BR IF OK
7654  050232  104027                          ERROR   27              ;REPORT REGISTER ERROR
7655  050234  000240                  67$:    NOP
7656
7657                                          ;RELEASE THE DRIVE FROM PORT B
7658
7659  050236  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
7660  050244  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
7661  050252  012760  000013  000000          MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
7662
7663                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
7664
7665  050260  005037  001250                  CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
7666  050264  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
7667  050272  060037  001122                  ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
7668  050276  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
7669  050304  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
7670  050312  016037  000012  001170          MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
7671  050320  013737  001170  001164          MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
7672  050326  042737  100100  001164          BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7673  050334  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
7674  050342  016037  000012  001172          MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
7675  050350  013737  001172  001166          MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
7676  050356  042737  100100  001166          BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
7677  050364  023737  001164  001166          CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
7678  050372  001006                          BNE     68$             ;BR IF NOT
7679  050374  005737  001164                  TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
7680  050400  001045                          BNE     70$             ;BR IF NOT
7681  050402  104046                          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
7682  050404  000137  050570                  JMP     72$             ;BYPASS THE REST OF THE CHECKS
7683  050410  013737  001170  001126  68$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
7684  050416  013737  001226  001234          MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
 7685  050424  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 7686  050432  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 7687  050436  001414                          BEQ     69$             ;BR IF ZERO
 7688  050440  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 7689  050446  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 7690  050454  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 7691  050462  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 7692  050466  001012                          BNE     70$             ;BR IF NOT
 7693  050470  012737  177777  001250  69$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 7694  050476  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 7695  050504  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 7696  050512  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 7697  050514  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 7698  050522  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 7699  050530  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 7700  050536  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
 7701  050540  104007                          ERROR   7               ;REPORT ERROR
 7702  050542  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 7703  050550  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 7704  050556  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 7705  050564  001401                          BEQ     72$             ;BR IF OK
 7706  050566  104007                          ERROR   7               ;REPORT ERROR
 7707  050570  000240                  72$:    NOP
 7708  050572  000004                          SCOPE                   ;LOOP ?
 7709  050574  000137  051132                  JMP     TST42           ;GO TO THE NEXT TEST
 7710
 7711                                          ;CHECK THE REGISTERS ON THE SELECTED PORT
 7712
 7713  050600                          TST41B:
 7714  050600  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7715  050604  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
 7716  050612  012737  000014  001122          MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7717  050620  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7718  050624  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7719  050630  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7720  050636  001403                          BEQ     64$             ;BR IF OK
 7721  050640  104006                          ERROR   6               ;TYPE MESSAGE 6
 7722  050642  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7723  050646  016037  000000  001126  64$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 7724  050654  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7725  050662  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7726  050666  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7727  050674  001404                          BEQ     65$             ;BR IF NOT
 7728  050676  104011                          ERROR   11              ;REPORT THE ERROR
 7729  050700  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7730  050706  000240                  65$:    NOP
 7731  050710  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7732  050714  016037  000040  001126          MOV     RPER2(R0),$BDDAT  ;GET CONTENTS OF RPER2
 7733  050722  012737  000040  001122          MOV     #RPER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7734  050730  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7735  050734  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7736  050740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7737  050746  001403                          BEQ     66$             ;BR IF OK
 7738  050750  104006                          ERROR   6               ;TYPE MESSAGE 6
 7739  050752  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7740  050756  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
```

```
 7741  050764  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7742  050772  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7743  050776  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7744  051004  001404                          BEQ     67$             ;BR IF NOT
 7745  051006  104011                          ERROR   11              ;REPORT THE ERROR
 7746  051010  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7747  051016  000240                  67$:    NOP
 7748  051020  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7749  051024  016037  000042  001126          MOV     RPER3(R0),$BDDAT  ;GET CONTENTS OF RPER3
 7750  051032  012737  000042  001122          MOV     #RPER3,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7751  051040  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7752  051044  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7753  051050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7754  051056  001403                          BEQ     68$             ;BR IF OK
 7755  051060  104006                          ERROR   6               ;TYPE MESSAGE 6
 7756  051062  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7757  051066  016037  000000  001126  68$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 7758  051074  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7759  051102  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7760  051106  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7761  051114  001404                          BEQ     69$             ;BR IF NOT
 7762  051116  104011                          ERROR   11              ;REPORT THE ERROR
 7763  051120  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7764  051126  000240                  69$:    NOP
 7765  051130  000207                          RTS     PC              ;RETURN
 7766
 7767                                  ;*****************************************************************
 7768                                  ;*TEST 42       TEST PORT INTERACTION FROM PORT 'B'
 7769                                  ;*
 7770                                  ;*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
 7771                                  ;*
 7772                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 7773                                  ;*
 7774                                  ;*  B.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 7775                                  ;*
 7776                                  ;*  C.  READ RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  VERIFY THAT PORT
 7777                                  ;*      'A' SEES 0'S FROM EACH OF THESE REGISTERS.
 7778                                  ;*
 7779                                  ;*  D.  CLEAR RPER1, RPER2, & RPER3 THROUGH PORT 'B'.
 7780                                  ;*
 7781                                  ;*  E.  WRITE 1'S INTO RPER1, RPER2, & RPER3 THROUGH PORT 'A'.  VERIFY THAT
 7782                                  ;*      PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
 7783                                  ;*
 7784                                  ;*  F.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE HAS
 7785                                  ;*      SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
 7786                                  ;*      SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
 7787                                  ;*
 7788                                  ;*  G.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
 7789                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 7790                                  ;*
 7791                                  ;*****************************************************************
 7792  051132                  TST42:
 7793  051132  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 7794  051136  001406                          BEQ     2$              ;BR IF NOT
 7795  051140  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 7796  051142  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
```

```
                                                                ;SET SINGLE TEST INDICATOR
                                                                ;TEST NUMBER
                                                                ;LOAD LOOP ON TEST ADDRESS
                                                                ;LOAD LOOP ON ERROR ADDRESS
                                                                ;;DO 4000. ITERATIONS
                                                                ;LOAD THE STACK POINTER

                                                                ;CLEAR ATTENTION BITS FOR BOTH PORTS

                                                                ;SELECT PORT #A
                                                                ;SEIZE THE DRIVE
                                                                ;ISSUE DRIVE CLEAR
                                                                ;RELEASE THE DRIVE
                                                                ;SELECT PORT #B
                                                                ;SEIZE THE DRIVE THROUGH PORT 'B'
                                                                ;ISSUE DRIVE CLEAR
                                                                ;RELEASE THE DRIVE

                                                                ;SEIZE THE DRIVE THROUGH PORT B

                                                                ;SELECT PORT B
                                                                ;STORE SEIZING PORT'S ADDRESS
                                                                ;WRITE RPDS1
                                                                ;'OPPOSITE' PORT ADDRESS
                                                                ;LOAD 1'S INTO RPER1 THROUGH PORT B
                                                                ;LOAD 1'S INTO RPER2 THROUGH PORT B
                                                                ;LOAD 1'S INTO RPER3 THROUGH PORT B
                                                                ;SELECT PORT A
                                                                ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                                ;CHECK THE REGISTERS THROUGH PORT A
                                                                ;SELECT PORT B
                                                                ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                                ;CLEAR RPER3 ON PORT B
                                                                ;CLEAR RPER2 ON PORT B
                                                                ;CLEAR RPER1 ON PORT B
                                                                ;CLEAR THE ATTENTION BIT FOR PORT B
                                                                ;SELECT PORT A
                                                                ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                                ;LOAD 1'S INTO RPER1 THROUGH PORT A
                                                                ;LOAD 1'S INTO RPER2 THROUGH PORT A
                                                                ;LOAD 1'S INTO RPER3 THROUGH PORT A
                                                                ;SELECT PORT B
                                                                ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                                ;CHECK THE REGISTERS THROUGH PORT B

                                                                ;RELEASE THE DRIVE FROM PORT B

                                                                ;SELECT PORT B
                                                                ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
                                                                ;ISSUE RELEASE THROUGH PORT B

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

                                                                ;CLEAR 'RELEASE ERROR' INDICATOR
                                                                ;COMPARISON CONSTANT
                                                                ;REGISTER ADDRESS INCREMENT
```

```
 7853  051532  060037  001122                    ADD     R0,$BDADR       ,REGISTER BASE ADDRESS FOR TYPEOUT
 7854  051536  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 7855  051544  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 7856  051552  016037  000012  001164            MOV     RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
 7857  051560  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 7858  051566  013737  001226  001234            MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 7859  051574  016037  000012  001126            MOV     RPDS1(R0),$BDDAT  ;DRIVE STATUS FROM PORT B
 7860  051602  001404                            BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
 7861  051604  005737  001164                    TST     $TMP0           ;IS STATUS FROM PORT A ZERO ?
 7862  051610  001401                            BEQ     66$             ;BR IF ZERO
 7863  051612  104031                            ERROR   31              ;REPORT DRIVE IN NEUTRAL
 7864  051614  013737  001164  001126  66$:     MOV     $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
 7865  051622  013737  001224  001234            MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
 7866  051630  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
 7867  051636  001401                            BEQ     67$             ;BR IF OK
 7868  051640  104027                            ERROR   27              ;REPORT REGISTER ERROR
 7869  051642  000240                    67$:    NOP
 7870
 7871                                            ;RELEASE THE DRIVE FROM PORT A
 7872
 7873  051644  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 7874  051652  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 7875  051660  012760  000013  000000            MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
 7876
 7877                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 7878
 7879  051666  005037  001250                    CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 7880  051672  012737  000012  001122            MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 7881  051700  060037  001122                    ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
 7882  051704  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 7883  051712  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 7884  051720  016037  000012  001170            MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 7885  051726  013737  001170  001164            MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
 7886  051734  042737  100100  001164            BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 7887  051742  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 7888  051750  016037  000012  001172            MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 7889  051756  013737  001172  001166            MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
 7890  051764  042737  100100  001166            BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 7891  051772  023737  001164  001166            CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 7892  052000  001006                            BNE     68$             ;BR IF NOT
 7893  052002  005737  001164                    TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 7894  052006  001045                            BNE     70$             ;BR IF NOT
 7895  052010  104046                            ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 7896  052012  000137  052176                    JMP     72$             ;BYPASS THE REST OF THE CHECKS
 7897  052016  013737  001170  001126  68$:     MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 7898  052024  013737  001226  001234            MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 7899  052032  113760  001226  000010            MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 7900  052040  005737  001164                    TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 7901  052044  001414                            BEQ     69$             ;BR IF ZERO
 7902  052046  013737  001224  001234            MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 7903  052054  013737  001172  001126            MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 7904  052062  113760  001224  000010            MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 7905  052070  005737  001166                    TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 7906  052074  001012                            BNE     70$             ;BR IF NOT
 7907  052076  012737  177777  001250  69$:     MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 7908  052104  012760  000011  000000            MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
```

```
 7909  052112  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 7910  052120  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 7911  052122  013737  001170  001126  70$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 7912  052130  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 7913  052136  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 7914  052144  001401                          BEQ     71$             ;BR IF OK FROM PORT A.
 7915  052146  104007                          ERROR   7               ;REPORT ERROR
 7916  052150  013737  001172  001126  71$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 7917  052156  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 7918  052164  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 7919  052172  001401                          BEQ     72$             ;BR IF OK
 7920  052174  104007                          ERROR   7               ;REPORT ERROR
 7921  052176  000240                  72$:    NOP
 7922  052200  000004                          SCOPE                   ;LOOP ?
 7923  052202  000137  052540                  JMP     TST43           ;GO TO THE NEXT TEST
 7924
 7925                                          ;CHECK THE REGISTERS ON THE SELECTED PORT
 7926
 7927  052206                          TST42B:
 7928  052206  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7929  052212  016037  000014  001126          MOV     RPER1(R0),$BDDAT  ;GET CONTENTS OF RPER1
 7930  052220  012737  000014  001122          MOV     #RPER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7931  052226  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7932  052232  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7933  052236  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7934  052244  001403                          BEQ     64$             ;BR IF OK
 7935  052246  104006                          ERROR   6               ;TYPE MESSAGE 6
 7936  052250  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7937  052254  016037  000000  001126  64$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 7938  052262  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7939  052270  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7940  052274  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7941  052302  001404                          BEQ     65$             ;BR IF NOT
 7942  052304  104011                          ERROR   11              ;REPORT THE ERROR
 7943  052306  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7944  052314  000240                  65$:    NOP
 7945  052316  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7946  052322  016037  000040  001126          MOV     RPER2(R0),$BDDAT  ;GET CONTENTS OF RPER2
 7947  052330  012737  000040  001122          MOV     #RPER2,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 7948  052336  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7949  052342  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7950  052346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7951  052354  001403                          BEQ     66$             ;BR IF OK
 7952  052356  104006                          ERROR   6               ;TYPE MESSAGE 6
 7953  052360  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7954  052364  016037  000000  001126  66$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 7955  052372  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7956  052400  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7957  052404  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7958  052412  001404                          BEQ     67$             ;BR IF NOT
 7959  052414  104011                          ERROR   11              ;REPORT THE ERROR
 7960  052416  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7961  052424  000240                  67$:    NOP
 7962  052426  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 7963  052432  016037  000042  001126          MOV     RPER3(R0),$BDDAT  ;GET CONTENTS OF RPER3
 7964  052440  012737  000042  001122          MOV     #RPER3,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
```

```
 7965  052446  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 7966  052452  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 7967  052456  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 7968  052464  001403                          BEQ     68$             ;BR IF OK
 7969  052466  104006                          ERROR   6               ;TYPE MESSAGE 6
 7970  052470  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
 7971  052474  016037  000000  001126  68$:    MOV     RPCS1(R0),$BDDAT  ;GET THE CONTENTS OF RHCS1
 7972  052502  012737  000000  001122          MOV     #RPCS1,$BDADR   ;FORM ADDRESS OF REGISTER
 7973  052510  060037  001122                  ADD     R0,$BDADR       ;ADDRESS BASE
 7974  052514  032737  020000  001126          BIT     #MCPE,$BDDAT    ;IS 'MCPE' SET ?
 7975  052522  001404                          BEQ     69$             ;BR IF NOT
 7976  052524  104011                          ERROR   11              ;REPORT THE ERROR
 7977  052526  012760  040000  000000          MOV     #TRE,RPCS1(R0)  ;CLEAR 'MCPE'
 7978  052534  000240                  69$:    NOP
 7979  052536  000207                          RTS     PC              ;RETURN
 7980
 7981
 7982                                  ;*****************************************************************
 7983                                  ;*TEST 43       TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
 7984                                  ;*
 7985                                  ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
 7986                                  ;*
 7987                                  ;*  A.  SET THE ATTENTION BIT FOR PORT 'A'.
 7988                                  ;*
 7989                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 7990                                  ;*
 7991                                  ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
 7992                                  ;*      FOR THE DRIVE IS SET.
 7993                                  ;*
 7994                                  ;*****************************************************************
 7995  052540                  TST43:
 7996  052540  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 7997  052544  001406                          BEQ     2$              ;BR IF NOT
 7998  052546  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 7999  052550  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8000  052554  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 8001  052562  112737  000043  001102  2$:     MOVB    #43,$TSTNM      ;TEST NUMBER
 8002  052570  012737  052612  001106          MOV     #TEST43,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8003  052576  012737  052612  001110          MOV     #TEST43,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 8004  052604  012737  000031  001176          MOV     #25.,$TIMES     ;;DO 25. ITERATIONS
 8005  052612  012706  001100          TEST43: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 8006
 8007                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 8008
 8009  052616  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 8010  052624  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 8011  052630  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8012  052636  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8013  052644  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 8014  052652  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 8015  052656  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8016  052664  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8017  052672  113760  001224  000010          MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
 8018  052700  012760  177777  000014          MOV     #-1,RPER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT A
 8019  052706  005060  000014                  CLR     RPER1(R0)       ;CLEAR THE ERRORS
 8020  052712  113760  001226  000010          MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
```

```
 8021  052720  005760  000012          1$:     TST     RPDS1(R0)         ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
 8022  052724  001775                          BEQ     1$                ;BR IF STILL SEIZED BY PORT A
 8023  052726  012737  000016  001122          MOV     #RPAS,$BDADR      ;FORM ADDRESS OF ATTN REG IF ERROR
 8024  052734  060037  001122                  ADD     R0,$BDADR         ;ADD THE ADDRESS BASE
 8025  052740  013737  001232  001124          MOV     ASR1,$GDDAT       ;GOOD DATA FOR ERROR MESSAGE
 8026  052746  013737  001232  001166          MOV     ASR1,$TMP1        ;MAKE DATA COMPARE MASK
 8027  052754  005137  001166                  COM     $TMP1             ;COMPLEMENT IT
 8028  052760  012737  053014  001110          MOV     #2$,$LPERR        ;LOAD LOOP ON ERROR ADDRESS
 8029  052766  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B
 8030  052774  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8031  053002  013737  001226  001236          MOV     PORTB,SEIZPT      ;'SEIZED' PORT ADDRESS
 8032  053010  005060  000012                  CLR     RPDS1(R0)         ;SEIZE THE DRIVE THROUGH PORT B
 8033  053014  016037  000016  001126  2$:     MOV     RPAS(R0),$BDDAT   ;GET THE CONTENTS OF THE ATTENTION REG
 8034  053022  013737  001126  001164          MOV     $BDDAT,$TMP0      ;PUT CONTENTS INTO WORKING LOCATION
 8035  053030  043737  001166  001164          BIC     $TMP1,$TMP0       ;CLEAR OTHER BITS
 8036  053036  023737  001124  001164          CMP     $GDDAT,$TMP0      ;SEE IF ATTN BIT FOR DRIVE SET
 8037  053044  001401                          BEQ     3$                ;BR IF SET
 8038  053046  104053                          ERROR   53                ;REPORT THE ERROR
 8039  053050                          3$:
 8040
 8041                                          ;RELEASE THE DRIVE FROM PORT B
 8042
 8043  053050  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B
 8044  053056  013737  001226  001234          MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8045  053064  012760  000013  000000          MOV     #13,RPCS1(R0)     ;ISSUE RELEASE THROUGH PORT B
 8046
 8047                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
 8048
 8049  053072  005037  001250                  CLR     RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 8050  053076  012737  000012  001122          MOV     #RPDS1,$BDADR     ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
 8051  053104  060037  001122                  ADD     R0,$BDADR         ;ADD THE I/O BASE ADDRESS
 8052  053110  012737  011700  001124          MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
 8053  053116  113760  001224  000010          MOVB    PORTA,RPCS2(R0)   ;SELECT PORT A.
 8054  053124  016037  000012  001170          MOV     RPDS1(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 8055  053132  013737  001170  001164          MOV     $TMP2,$TMP0       ;COPY IT INTO '$TMP0'
 8056  053140  042737  100100  001164          BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 8057  053146  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B.
 8058  053154  016037  000012  001172          MOV     RPDS1(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 8059  053162  013737  001172  001166          MOV     $TMP3,$TMP1       ;COPY IT INTO '$TMP1'
 8060  053170  042737  100100  001166          BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 8061  053176  023737  001164  001166          CMP     $TMP0,$TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 8062  053204  001006                          BNE     64$               ;BR IF NOT
 8063  053206  005737  001164                  TST     $TMP0             ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 8064  053212  001045                          BNE     66$               ;BR IF NOT
 8065  053214  104046                          ERROR   46                ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
 8066  053216  000137  053416                  JMP     68$               ;BYPASS THE REST OF THE CHECKS
 8067  053222  013737  001170  001126  64$:     MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 8068  053230  013737  001226  001234          MOV     PORTB,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 8069  053236  113760  001226  000010          MOVB    PORTB,RPCS2(R0)   ;SELECT PORT B.
 8070  053244  005737  001164                  TST     $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
 8071  053250  001414                          BEQ     65$               ;BR IF ZERO
 8072  053252  013737  001224  001234          MOV     PORTA,PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 8073  053260  013737  001172  001126          MOV     $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
 8074  053266  113760  001224  000010          MOVB    PORTA,RPCS2(R0)   ;SELECT PORT A.
 8075  053274  005737  001166                  TST     $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
 8076  053300  001012                          BNE     66$               ;BR IF NOT
```

```
 8077  053302  012737  177777  001250  65$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 8078  053310  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 8079  053316  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8080  053324  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 8081  053326  013737  001170  001126  66$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 8082  053334  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8083  053342  042737  100000  001170          BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
 8084  053350  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8085  053356  001401                          BEQ     67$             ;BR IF OK FROM PORT A.
 8086  053360  104007                          ERROR   7               ;REPORT ERROR
 8087  053362  013737  001172  001126  67$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 8088  053370  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8089  053376  042737  100000  001172          BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
 8090  053404  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8091  053412  001401                          BEQ     68$             ;BR IF OK
 8092  053414  104007                          ERROR   7               ;REPORT ERROR
 8093  053416  000240                  68$:    NOP
 8094  053420  000004                          SCOPE                   ;LOOP ?
 8095
 8096                                  ;;*****************************************************************
 8097                                  ;*TEST 44       TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
 8098                                  ;*
 8099                                  ;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
 8100                                  ;*
 8101                                  ;*  A.  SET THE ATTENTION BIT FOR PORT 'B'.
 8102                                  ;*
 8103                                  ;*  B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 8104                                  ;*
 8105                                  ;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
 8106                                  ;*      FOR THE DRIVE IS SET.
 8107                                  ;*
 8108                                  ;;*****************************************************************
 8109  053422                          ;ST44:
 8110  053422  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 8111  053426  001406                          BEQ     2$              ;BR IF NOT
 8112  053430  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 8113  053432  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8114  053436  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 8115  053444  112737  000044  001102  2$:     MOVB    #44,$TSTNM      ;TEST NUMBER
 8116  053452  012737  053474  001106          MOV     #TEST44,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8117  053460  012737  053474  001110          MOV     #TEST44,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 8118  053466  012737  000031  001176          MOV     #25.,$TIMES     ;;DO 25. ITERATIONS
 8119  053474  012706  001100          TEST44: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 8120
 8121                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 8122
 8123  053500  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 8124  053506  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 8125  053512  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8126  053520  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8127  053526  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 8128  053534  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 8129  053540  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8130  053546  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8131  053554  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 8132  053562  012760  177777  000014          MOV     #-1,RPER1(R0)   ;SET ERRORS TO FORCE ATTN BIT ON PORT B
```

```
8133  053570  005060  000014                    CLR     RPER1(R0)          ;CLEAR THE ERRORS
8134  053574  113760  001224  000010            MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A
8135  053602  005760  000012          1$:       TST     RPDS1(R0)          ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
8136  053606  001775                            BEQ     1$                 ;BR IF STILL SEIZED BY PORT B
8137  053610  012737  000016  001122            MOV     #RPAS,$BDADR       ;FORM ADDRESS OF ATTN REG IF ERROR
8138  053616  060037  001122                    ADD     R0,$BDADR          ;ADD THE ADDRESS BASE
8139  053622  013737  001232  001124            MOV     ASR1,$GDDAT        ;GOOD DATA FOR ERROR MESSAGE
8140  053630  013737  001232  001166            MOV     ASR1,$TMP1         ;MAKE DATA COMPARE MASK
8141  053636  005137  001166                    COM     $TMP1              ;COMPLEMENT IT
8142  053642  012737  053676  001110            MOV     #2$,$LPERR         ;LOAD LOOP ON ERROR ADDRESS
8143  053650  113760  001224  000010            MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A
8144  053656  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8145  053664  013737  001224  001236            MOV     PORTA,SEIZPT       ;'SEIZED' PORT ADDRESS
8146  053672  005060  000012                    CLR     RPDS1(R0)          ;SEIZE THE DRIVE THROUGH PORT A
8147  053676  016037  000016  001126  2$:       MOV     RPAS(R0),$BDDAT    ;GET THE CONTENTS OF THE ATTENTION REG
8148  053704  013737  001126  001164            MOV     $BDDAT,$TMP0       ;PUT CONTENTS INTO WORKING LOCATION
8149  053712  043737  001166  001164            BIC     $TMP1,$TMP0        ;CLEAR OTHER BITS
8150  053720  023737  001124  001164            CMP     $GDDAT,$TMP0       ;SEE IF ATTN BIT FOR DRIVE SET
8151  053726  001401                            BEQ     3$                 ;BR IF SET
8152  053730  104053                            ERROR   53                 ;REPORT THE ERROR
8153  053732                          3$:
8154
8155                                            ;RELEASE THE DRIVE FROM PORT A
8156
8157  053732  113760  001224  000010            MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A
8158  053740  013737  001224  001234            MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8159  053746  012760  000013  000000            MOV     #13,RPCS1(R0)      ;ISSUE RELEASE THROUGH PORT A
8160
8161                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8162
8163  053754  005037  001250                    CLR     RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8164  053760  012737  000012  001122            MOV     #RPDS1,$BDADR      ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8165  053766  060037  001122                    ADD     R0,$BDADR          ;ADD THE I/O BASE ADDRESS
8166  053772  012737  011700  001124            MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8167  054000  113760  001224  000010            MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A.
8168  054006  016037  000012  001170            MOV     RPDS1(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8169  054014  013737  001170  001164            MOV     $TMP2,$TMP0        ;COPY IT INTO '$TMP0'
8170  054022  042737  100100  001164            BIC     #ATA!VV,$TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8171  054030  113760  001226  000010            MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B.
8172  054036  016037  000012  001172            MOV     RPDS1(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8173  054044  013737  001172  001166            MOV     $TMP3,$TMP1        ;COPY IT INTO '$TMP1'
8174  054052  042737  100100  001166            BIC     #ATA!VV,$TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8175  054060  023737  001164  001166            CMP     $TMP0,$TMP1        ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8176  054066  001006                            BNE     64$                ;BR IF NOT
8177  054070  005737  001164                    TST     $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8178  054074  001045                            BNE     66$                ;BR IF NOT
8179  054076  104046                            ERROR   46                 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8180  054100  000137  054300                    JMP     68$                ;BYPASS THE REST OF THE CHECKS
8181  054104  013737  001170  001126  64$:       MOV     $TMP2,$BDDAT       ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8182  054112  013737  001226  001234            MOV     PORTB,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8183  054120  113760  001226  000010            MOVB    PORTB,RPCS2(R0)    ;SELECT PORT B.
8184  054126  005737  001164                    TST     $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
8185  054132  001414                            BEQ     65$                ;BR IF ZERO
8186  054134  013737  001224  001234            MOV     PORTA,PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8187  054142  013737  001172  001126            MOV     $TMP3,$BDDAT       ;'BAD DATA' FOR ERROR TYPE OUT
8188  054150  113760  001224  000010            MOVB    PORTA,RPCS2(R0)    ;SELECT PORT A.
```

```
 8189  054156  005737  001166                 TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 8190  054162  001012                         BNE     66$             ;BR IF NOT
 8191  054164  012737  177777  001250  65$:   MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 8192  054172  012760  000011  000000         MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 8193  054200  012760  000013  000000         MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8194  054206  104026                         ERROR   26              ;TYPE ERROR MESSAGE 26
 8195  054210  013737  001170  001126  66$:   MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 8196  054216  013737  001224  001234         MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8197  054224  042737  100000  001170         BIC     #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
 8198  054232  023737  001124  001170         CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8199  054240  001401                         BEQ     67$             ;BR IF OK FROM PORT A.
 8200  054242  104007                         ERROR   7               ;REPORT ERROR
 8201  054244  013737  001172  001126  67$:   MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 9202  054252  013737  001226  001234         MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8203  054260  042737  100000  001172         BIC     #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT
 8204  054266  023737  001124  001172         CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8205  054274  001401                         BEQ     68$             ;BR IF OK
 8206  054276  104007                         ERROR   7               ;REPORT ERROR
 8207  054300  000240                 68$:    NOP
 8208  054302  000004                         SCOPE                   ;LOOP ?
 8209  054304  000137  056224                 JMP     $EOP            ;GO TO END OF TEST
 8210
 8211                          ;;*****************************************************************
 8212
 8213                          .SBTTL  *** SPECIAL TESTS FOR THE M7775 ('DP') BOARD ***
 8214
 8215                          ;;*****************************************************************
 8216
 8217
 8218                          ;;*****************************************************************
 8219                          ;*TEST 45       TEST NO TIMEOUT THROUGH PORT 'A'
 8220                          ;*
 8221                          ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
 8222                          ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
 8223                          ;*
 8224                          ;*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
 8225                          ;*
 8226                          ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
 8227                          ;*
 8228                          ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B'.  VERIFY THAT THE DRIVE
 8229                          ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
 8230                          ;*      SET FOR PORT 'B'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
 8231                          ;*
 8232                          ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
 8233                          ;*      BEEN RELEASED.
 8234                          ;*
 8235                          ;*  E.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
 8236                          ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 8237                          ;*
 8238                          ;*****************************************************************
 8239  054310                  ST45:
 8240  054310  005737  001274                 TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
 8241  054314  001406                         BEQ     2$              ;BR IF NOT
 8242  054316  100002                         BPL     1$              ;BR IF JUST ENTERED TEST
 8243  054320  000137  002602                 JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8244  054324  012737  177777  001274  1$:    MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
```

```
 8245  054332  112737  000045  001102  2$:     MOVB    #45,$TSTNM      ;TEST NUMBER
 8246  054340  012737  054362  001106          MOV     #TEST45,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8247  054346  012737  054362  001110          MOV     #TEST45,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 8248  054354  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
 8249  054362  012706  001100          TEST45: MOV     #STACK,SP       ;LOAD THE STACK POINTER
 8250
 8251                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
 8252
 8253  054366  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
 8254  054374  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
 8255  054400  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8256  054406  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8257  054414  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
 8258  054422  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
 8259  054426  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
 8260  054434  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8261
 8262                                          ;;*****************************************************************
 8263
 8264                                          ;SEIZE THE DRIVE THROUGH PORT B
 8265
 8266  054442  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 8267  054450  013737  001226  001236          MOV     PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
 8268  054456  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
 8269  054462  013737  001224  001240          MOV     PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
 8270  054470  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
 8271  054476  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8272
 8273                                          ;;*****************************************************************
 8274                                          ;SET REQUEST THROUGH PORT A
 8275
 8276  054504  005060  000012                  CLR     RPDS1(R0)       ;SET REQUEST FOR PORT A
 8277  054510  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
 8278  054516  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 8279
 8280                                          ;;*****************************************************************
 8281                                          ;RELEASE THE DRIVE THROUGH PORT B
 8282
 8283  054524  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE DRIVE THROUGH PORT B
 8284
 8285                                          ;;*****************************************************************
 8286                                          ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
 8287
 8288  054532  013737  001260  001254          MOV     TIMEAP,WATCH    ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%.
 8289
 8290                                          ;;*****************************************************************
 8291                                          ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
 8292
 8293  054540  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
 8294  054544  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
 8295  054552  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 8296  054560  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
 8297  054564  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
 8298  054570  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
 8299  054576  001403                          BEQ     66$             ;BR IF OK
 8300  054600  104031                          ERROR   31              ;TYPE MESSAGE 31
```

```
8301   054602  005137  001244           66$:    COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8302   054606  000240                           NOP
8303   054610  005737  001244                   TST     CKERR           ;REGISTER OK ?
8304   054614  001402                           BEQ     .+6             ;BR IF OK
8305   054616  000137  055254                   JMP     1$              ;BYPASS REST OF TEST IF NOT
8306   054622  005737  001254                   TST     WATCH           ;WATCH EQUAL ZERO ?
8307   054626  001375                           BNE     .-4             ;BR IF NOT
8308
8309                                    ;;*****************************************************************
8310                                    ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
8311
8312   054630  013737  001224  001234           MOV     PORTA,PTNBR     ;PORT NUMBER FOR TYPEOUT
8313   054636  005037  001244                   CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8314   054642  016037  000012  001126           MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8315   054650  012737  000012  001122           MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8316   054656  060037  001122                   ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8317   054662  005037  001124                   CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8318   054666  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8319   054674  001403                           BEQ     68$             ;BR IF OK
8320   054676  104035                           ERROR   35              ;TYPE MESSAGE 35
8321   054700  005137  001244                   COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8322   054704  000240                   68$:    NOP
8323   054706  005737  001244                   TST     CKERR           ;REGISTER OK ?
8324   054712  001402                           BEQ     .+6             ;BR IF OK
8325   054714  000137  055254                   JMP     1$              ;BYPASS REST OF TEST IF NOT
8326
8327                                    ;;*****************************************************************
8328
8329                                            ;RELEASE THE DRIVE FROM PORT A
8330
8331   054720  113760  001224  000010           MOVB    PORTA,RPCS2(R0)  ;SELECT PORT A
8332   054726  013737  001224  001234           MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8333   054734  012760  000013  000000           MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
8334
8335                                            ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8336
8337   054742  005037  001250                   CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8338   054746  012737  000012  001122           MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8339   054754  060037  001122                   ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8340   054760  012737  011700  001124           MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8341   054766  113760  001224  000010           MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8342   054774  016037  000012  001170           MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8343   055002  013737  001170  001164           MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8344   055010  042737  100100  001164           BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8345   055016  113760  001226  000010           MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8346   055024  016037  000012  001172           MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8347   055032  013737  001172  001166           MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8348   055040  042737  100100  001166           BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8349   055046  023737  001164  001166           CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8350   055054  001006                           BNE     70$             ;BR IF NOT
8351   055056  005737  001164                   TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8352   055062  001045                           BNE     72$             ;BR IF NOT
8353   055064  104046                           ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8354   055066  000137  055252                   JMP     74$             ;BYPASS THE REST OF THE CHECKS
8355   055072  013737  001170  001126   70$:    MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8356   055100  013737  001226  001234           MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
```

```
 8357  055106  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
 8358  055114  005737  001164                  TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
 8359  055120  001414                          BEQ     71$             ;BR IF ZERO
 8360  055122  013737  001224  001234          MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 8361  055130  013737  001172  001126          MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
 8362  055136  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
 8363  055144  005737  001166                  TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
 8364  055150  001012                          BNE     72$             ;BR IF NOT
 8365  055152  012737  177777  001250  71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
 8366  055160  012760  000011  000000          MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
 8367  055166  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
 8368  055174  104026                          ERROR   26              ;TYPE ERROR MESSAGE 26
 8369  055176  013737  001170  001126  72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
 8370  055204  013737  001224  001234          MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
 8371  055212  023737  001124  001170          CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
 8372  055220  001401                          BEQ     73$             ;BR IF OK FROM PORT A.
 8373  055222  104007                          ERROR   7               ;REPORT ERROR
 8374  055224  013737  001172  001126  73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
 8375  055232  013737  001226  001234          MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
 8376  055240  023737  001124  001172          CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
 8377  055246  001401                          BEQ     74$             ;BR IF OK
 8378  055250  104007                          ERROR   7               ;REPORT ERROR
 8379  055252  000240                  74$:    NOP
 8380
 8381  055254  000004                  1$:     SCOPE                   ;LOOP ?
 8382
 8383                                  ;;*****************************************************************
 8384                                  ;*TEST 46       TEST NO TIMEOUT THROUGH PORT 'B'
 8385                                  ;*
 8386                                  ;*VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
 8387                                  ;*      SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
 8388                                  ;*
 8389                                  ;*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
 8390                                  ;*
 8391                                  ;*  B.  SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
 8392                                  ;*
 8393                                  ;*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A'.  VERIFY THAT THE DRIVE
 8394                                  ;*      HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
 8395                                  ;*      SET FOR PORT 'A'.  REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
 8396                                  ;*
 8397                                  ;*  D.  WAIT THE TIMEOUT INTERVAL + 25%.  VERIFY THAT THE DRIVE HAS NOT
 8398                                  ;*      BEEN RELEASED.
 8399                                  ;*
 8400                                  ;*  E.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
 8401                                  ;*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
 8402                                  ;*
 8403                                  ;*****************************************************************
 8404  055256                  TST46:
 8405  055256  005737  001274                  TST     KYBCTL          ;PERFORMING ONLY SINGLE TESTS ^
 8406  055262  001406                          BEQ     2$              ;BR IF NOT
 8407  055264  100002                          BPL     1$              ;BR IF JUST ENTERED TEST
 8408  055266  000137  002602                  JMP     EXEC            ;RETURN & GET NEXT TEST NUMBER
 8409  055272  012737  177777  001274  1$:     MOV     #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
 8410  055300  112737  000046  001102  2$:     MOVB    #46,$TSTNM      ;TEST NUMBER
 8411  055306  012737  055330  001106          MOV     #TEST46,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 8412  055314  012737  055330  001110          MOV     #TEST46,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
```

```
8413  055322  012737  000004  001176          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
8414  055330  012706  001100          TEST46: MOV     #STACK,SP       ;LOAD THE STACK POINTER
8415
8416                                          ;CLEAR ATTENTION BITS FOR BOTH PORTS
8417
8418  055334  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT #A
8419  055342  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE
8420  055346  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
8421  055354  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8422  055362  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT #B
8423  055370  005060  000012                  CLR     RPDS1(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
8424  055374  012760  000011  000000          MOV     #11,RPCS1(R0)   ;ISSUE DRIVE CLEAR
8425  055402  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
8426
8427                                  ;;**********************************************************************
8428
8429                                          ;SEIZE THE DRIVE THROUGH PORT A
8430
8431  055410  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
8432  055416  013737  001224  001236          MOV     PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
8433  055424  005060  000012                  CLR     RPDS1(R0)       ;WRITE RPDS1
8434  055430  013737  001226  001240          MOV     PORTB,OPPRT     ;'OPPOSITE' PORT ADDRESS
8435  055436  113760  001226  000010          MOVB    PORTB,RPCS2(R0) ;SELECT PORT B
8436  055444  013737  001226  001234          MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8437
8438                                  ;;**********************************************************************
8439                                  ;SET REQUEST THROUGH PORT B
8440
8441  055452  005060  000012                  CLR     RPDS1(R0)       ;SET REQUEST FOR PORT B
8442  055456  113760  001224  000010          MOVB    PORTA,RPCS2(R0) ;SELECT PORT A
8443  055464  013737  001224  001234          MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8444
8445                                  ;;**********************************************************************
8446                                  ;RELEASE THE DRIVE THROUGH PORT A
8447
8448  055472  012760  000013  000000          MOV     #13,RPCS1(R0)   ;RELEASE DRIVE THROUGH PORT A
8449
8450                                  ;;**********************************************************************
8451                                  ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
8452
8453  055500  013737  001266  001254          MOV     TIMEBP,WATCH    ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
8454
8455                                  ;;**********************************************************************
8456                                  ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
8457
8458  055506  005037  001244                  CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8459  055512  016037  000012  001126          MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8460  055520  012737  000012  001122          MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8461  055526  060037  001122                  ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8462  055532  005037  001124                  CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8463  055536  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8464  055544  001403                          BEQ     66$             ;BR IF OK
8465  055546  104031                          ERROR   31              ;TYPE MESSAGE 31
8466  055550  005137  001244                  COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8467  055554  000240                  66$:    NOP
8468  055556  005737  001244                  TST     CKERR           ;REGISTER OK ?
```

```
8469  055562  001402                        BEQ     .+6             ;BR IF OK
8470  055564  000137  056222                JMP     I$              ;BYPASS REST OF TEST IF NOT
8471  055570  005737  001254                TST     WATCH           ;WATCH EQUAL ZERO ?
8472  055574  001375                        BNE     .-4             ;BR IF NOT
8473
8474                                  ;;*****************************************************************
8475                                  ;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT
8476
8477  055576  013737  001226  001234        MOV     PORTB,PTNBR     ;PORT NUMBER FOR TYPEOUT
8478  055604  005037  001244                CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
8479  055610  016037  000012  001126        MOV     RPDS1(R0),$BDDAT  ;GET CONTENTS OF RPDS1
8480  055616  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
8481  055624  060037  001122                ADD     R0,$BDADR       ;ADD RH11 BASE ADDRESS
8482  055630  005037  001124                CLR     $GDDAT          ;WHAT REGISTER SHOULD BE
8483  055634  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
8484  055642  001403                        BEQ     68$             ;BR IF OK
8485  055644  104035                        ERROR   35              ;TYPE MESSAGE 35
8486  055646  005137  001244                COM     CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
8487  055652  000240                68$:    NOP
8488  055654  005737  001244                TST     CKERR           ;REGISTER OK ?
8489  055660  001402                        BEQ     .+6             ;BR IF OK
8490  055662  000137  056222                JMP     I$              ;BYPASS REST OF TEST IF NOT
8491
8492                                  ;;*****************************************************************
8493
8494                                          ;RELEASE THE DRIVE FROM PORT B
8495
8496  055666  113760  001226  000010        MOVB    PORTB,RPCS2(R0)  ;SELECT PORT B
8497  055674  013737  001226  001234        MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
8498  055702  012760  000013  000000        MOV     #13,RPCS1(R0)   ;ISSUE RELEASE THROUGH PORT B
8499
8500                                          ;VERIFY THAT THE DRIVE IS IN NEUTRAL
8501
8502  055710  005037  001250                CLR     RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
8503  055714  012737  000012  001122        MOV     #RPDS1,$BDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
8504  055722  060037  001122                ADD     R0,$BDADR       ;ADD THE I/O BASE ADDRESS
8505  055726  012737  011700  001124        MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT  ;COMPARISON CONSTANT
8506  055734  113760  001224  000010        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
8507  055742  016037  000012  001170        MOV     RPDS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
8508  055750  013737  001170  001164        MOV     $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
8509  055756  042737  100100  001164        BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8510  055764  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8511  055772  016037  000012  001172        MOV     RPDS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
8512  056000  013737  001172  001166        MOV     $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
8513  056006  042737  100100  001166        BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
8514  056014  023737  001164  001166        CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
8515  056022  001006                        BNE     70$             ;BR IF NOT
8516  056024  005737  001164                TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
8517  056030  001045                        BNE     72$             ;BR IF NOT
8518  056032  104046                        ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
8519  056034  000137  056220                JMP     74$             ;BYPASS THE REST OF THE CHECKS
8520  056040  013737  001170  001126  70$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
8521  056046  013737  001226  001234        MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
8522  056054  113760  001226  000010        MOVB    PORTB,RPCS2(R0) ;SELECT PORT B.
8523  056062  005737  001164                TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
8524  056066  001414                        BEQ     71$             ;BR IF ZERO
```

```
                                        MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
                                        MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
                                        MOVB    PORTA,RPCS2(R0) ;SELECT PORT A.
                                        TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
                                        BNE     72$             ;BR IF NOT
                                71$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
                                        MOV     #11,RPCS1(R0)   ;CLEAR THE DRIVE
                                        MOV     #13,RPCS1(R0)   ;RELEASE THE DRIVE
                                        ERROR   26              ;TYPE ERROR MESSAGE 26
                                72$:    MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
                                        MOV     PORTA,PTNBR     ;CHANGE PORT NUMBER
                                        CMP     $GDDAT,$TMP2    ;ALL BITS OK ?
                                        BEQ     73$             ;BR IF OK FROM PORT A.
                                        ERROR   7               ;REPORT ERROR
                                73$:    MOV     $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
                                        MOV     PORTB,PTNBR     ;CHANGE PORT NUMBER
                                        CMP     $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
                                        BEQ     74$             ;BR IF OK
                                        ERROR   7               ;REPORT ERROP
                                74$:    NOP

                                1$:     SCOPE                   ;LOOP ?

                                .SBTTL  END OF PASS ROUTINE

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

                                $EOP:
                                        TST     KYBCTL          ;ENTERED TEST VIA KEYBOARD COMMAND ?
                                        BEQ     .+6             ;BR IF NOT
                                        JMP     EXEC            ;RETURN TO KEYBOARD CONTROL
                                        CLR     $TSTNM          ;;ZERO THE TEST NUMBER
                                        CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
                                        INC     $PASS           ;;INCREMENT THE PASS NUMBER
                                        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
                                        DEC     (PC)+           ;;LOOP?
                                $EOPCT: .WORD   1
                                        BGT     $DOAGN          ;;YES
                                        MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
                                $ENDCT: .WORD   1
                                        $EOPCT
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
                                64$:
                                        MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
                                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
```

```
 8581                                   ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
 8582  056376                           66$:
 8583  056376  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
 8584                                                                   ;;TOTAL NUMBER OF ERRORS
 8585  056402  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8586  056404  104401  001207                  TYPE    $CRLF           ;;TYPE CARRIAGE RETURN, LINE FEED
 8587  056410  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
 8588  056414  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 8589  056420  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 8590  056422  000005                          RESET                   ;;CLEAR THE WORLD
 8591  056424  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 8592  056426  000240                          NOP                     ;;SAVE ROOM
 8593  056430  000240                          NOP                     ;;FOR
 8594  056432  000240                          NOP                     ;;ACT11
 8595  056434                          $DOAGN:
 8596  056434  000137                          JMP     @(PC)+          ;;RETURN
 8597  056436  003064                  $RTNAD: .WORD   TST1AA
 8598  056440     377       377       000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 8599  056444                                  .EVEN
 8600
 8601                                   ;;******************************************************************
 8602
 8603                                   .SBTTL  *** SUBROUTINES ***
 8604
 8605                                   ;;******************************************************************
 8606
 8607
 8608                                   ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
 8609                                   ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
 8610
 8611  056444  012737  056514  000004  CKCLK:  MOV     #CKCLK1,@#ERRVEC  ;SET UP VECTOR FOR CLOCK CHECK
 8612  056452  005037  000006                  CLR     @#ERRVEC+2        ;NEW PSW
 8613  056456  005777  122530                  TST     @$LKCSR         ;CHECK FOR KW11-P
 8614  056462  013701  001216                  MOV     $LPVEC,R1       ;KW11-P VECTOR ADDRESS
 8615  056466  012721  056576                  MOV     #CLOCK,(R1)+    ;SET UP KW11-P VECTOR
 8616  056472  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
 8617  056476  012777  177777  122510          MOV     #-1,@$LKCSB     ;LOAD COUNTER BUFFER WITH 1'S
 8618  056504  012777  000135  122500          MOV     #135,@$LKCSR    ;SET CLOCK - CNT UP, 16MS, CONT INT
 8619  056512  000425                          BR      CKCLK3
 8620  056514  062706  000004          CKCLK1: ADD     #4,SP           ;RESTORE THE STACK POINTER
 8621  056520  012737  056556  000004          MOV     #CKCLK2,@#ERRVEC  ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
 8622  056526  005777  122466                  TST     @$LKS           ;LOOK FOR KW11-L
 8623  056532  013701  001222                  MOV     $LLVEC,R1       ;KW11-L VECTOR ADDRESS
 8624  056536  012721  056576                  MOV     #CLOCK,(R1)+    ;SET UP KW11-L VECTOR
 8625  056542  012711  000300                  MOV     #300,(R1)       ;PSW - PRI 6
 8626  056546  012777  000100  122444          MOV     #100,@$LKS      ;SET KW11-L INTERRUPT
 8627  056554  000404                          BR      CKCLK3
 8628  056556  062706  000004          CKCLK2: ADD     #4,SP           ;RESTORE THE STACK POINTER
 8629  056562  062716  000002                  ADD     #2,(SP)         ;INCREMENT RETURN, NO CLOCK
 8630  056566  012737  000006  000004  CKCLK3: MOV     #6,@#ERRVEC     ;RESTORE THE ERROR VECTOR
 8631  056574  000207                          RTS     PC
 8632
 8633                                   ;ROUTINE TO COUNT CLOCK TICKS
 8634
 8635  056576  062737  000021  001252  CLOCK:  ADD     #17.,TIME       ;ADD 17 MS TO ELAPSED TIME COUNTER
 8636  056604  005737  001254                  TST     WATCH           ;IS WATCH ALREADY ZERO ?
```

```
8637  056610  001406                        BEQ     1$                      ;BR IF IT IS
8638  056612  162737  000021  001254        SUB     #17.,WATCH              ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
8639  056620  100002                        BPL     1$                      ;BR IF NOT MINUS
8640  056622  005037  001254                CLR     WATCH                   ;CLEAR WATCH DOG COUNTER
8641  056626  000002                1$:     RTI                             ;RETURN
8642
8643
8644
8645  056630  162706  000004        TOLER:  SUB     #4,SP   ;SETUP STACK
8646  056634  016616  000004                MOV     4(SP),(SP)              ;SAVE STACK
8647  056640  013546                        MOV     @(R5)+,-(SP)            ;GET TIME VALUE
8648  056642  011666  000004                MOV     (SP),4(SP)              ;MOVE TIME VALUE
8649  056646  011666  000006                MOV     (SP),6(SP)              ;MOVE VALUE AGAIN
8650  056652  006216                        ASR     (SP)                    ;DIVIDE BY 2
8651  056654  006216                        ASR     (SP)                    ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
8652  056656  061666  000004                ADD     (SP),4(SP)              ;CALCULATE UPPER LIMIT FOR TIMEOUT
8653  056662  162666  000004                SUB     (SP)+,4(SP)             ;CALCULATE LOWER LIMIT FOR TIMEOUT
8654  056666  000205                        RTS     R5                      ;RETURN WITH TOLERANCES ON THE STACK
8655
8656                                ;;**********************************************************************
8657
8658                                .SBTTL  'SYSMAC' UTILITY ROUTINES
8659
8660                                ;;**********************************************************************
8661
8662                                .SBTTL  SCOPE HANDLER ROUTINE
8663
8664                                ;;**********************************************************************
8665                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8666                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>
8667                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8668                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8669                                ;*SW14=1         LOOP ON TEST
8670                                ;*SW11=1         INHIBIT ITERATIONS
8671                                ;*SW09=1         LOOP ON ERROR
8672                                ;*CALL
8673                                ;*      SCOPE           ;;SCOPE=IOT
8674
8675  056670                        $SCOPE:
8676  056670  104407                        CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
8677  056672  032777  040000  122240  1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
8678  056700  001101                        BNE     $OVER                   ;;YES IF SW14=1
8679                                ;*****START OF CODE FOR THE XOR TESTER*****
8680  056702  000416                $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
8681                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240
8682  056704  013746  000004                MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
8683  056710  012737  056730  000004        MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
8684  056716  005737  177060                TST     @#177060                ;;TIME OUT ON XOR?
8685  056722  012637  000004                MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
8686  056726  000453                        BR      $SVLAD                  ;;GO TO THE NEXT TEST
8687  056730  022626                5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
8688  056732  012637  000004                MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
8689  056736  000413                        BR      7$                      ;;LOOP ON THE PRESENT TEST
8690  056740                        6$:;*****END OF CODE FOR THE XOR TESTER*****
8691  056740  105737  001103        2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
8692  056744  001421                        BEQ     3$                      ;;BR IF NO
```

```
 8693  056746  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 8694  056754  101015                          BHI     3$              ;;BR IF NO
 8695  056756  032777  001000  122154          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 8696  056764  001404                          BEQ     4$              ;;BR IF NO
 8697  056766  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 8698  056774  000443                          BR      $OVER
 8699  056776  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 8700  057002  005037  001176                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 8701  057006  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 8702  057010  032777  004000  122122  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 8703  057016  001011                          BNE     1$              ;;BR IF YES
 8704  057020  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 8705  057024  001406                          BEQ     1$              ;;     INHIBIT ITERATIONS
 8706  057026  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 8707  057032  023737  001176  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 8708  057040  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 8709  057042  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 8710  057050  013737  057120  001176          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 8711  057056  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 8712  057062  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 8713  057066  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 8714  057072  005037  001200                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 8715  057076  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 8716  057104  013777  001102  122030  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 8717  057112  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 8718  057116  000002                          RTI                     ;;FIXES PS
 8719  057120  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
 8720                          .SBTTL  ERROR HANDLER ROUTINE
 8721
 8722                          ;;*****************************************************************
 8723                          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 8724                          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 8725                          ;*AND GO TO $ERRTYP ON ERROR
 8726                          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 8727                          ;*SW15=1        HALT ON ERROR
 8728                          ;*SW13=1        INHIBIT ERROR TYPEOUTS
 8729                          ;*SW10=1        BELL ON ERROR
 8730                          ;*CALL
 8731                          ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 8732
 8733  057122                  $ERROR:
 8734  057122  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 8735  057124  113737  001102  001242          MOVB    $TSTNM,TSTNUM
 8736  057132  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 8737  057136  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 8738  057140  013777  001102  121774          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 8739  057146  032777  002000  121764          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 8740  057154  001402                          BEQ     1$              ;;NO - SKIP
 8741  057156  104401  001202                  TYPE    ,$BELL          ;;RING BELL
 8742  057162  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 8743  057166  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 8744  057172  162737  000002  001116          SUB     #2,$ERRPC
 8745  057200  117737  121712  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 8746  057206  032777  020000  121724          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 8747  057214  001004                          BNE     20$             ;;SKIP TYPEOUTS
 8748  057216  004737  057254                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
```

```
8749  057222  104401  001207            TYPE    ,$CRLF
8750  057226                    20$:
8751  057226  005777  121706    2$:     TST     @SWR            ;;HALT ON ERROR
8752  057232  100002                    BPL     3$              ;;SKIP IF CONTINUE
8753  057234  000000                    HALT                    ;;HALT ON ERROR!
8754  057236  104407                    CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
8755  057240                    3$:
8756  057240  022737  056424  000042    CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
8757  057246  001001                    BNE     6$              ;;BRANCH IF NO
8758  057250  000000                    HALT                    ;;YES
8759  057252                    6$:
8760  057252  000002                    RTI                     ;;RETURN
8761                            .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
8762
8763                            ;;**********************************************************
8764                            ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
8765                            ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
8766                            ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
8767
8768  057254                    $ERRTYP:
8769  057254  104401  001207            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8770  057260  010046                    MOV     R0,-(SP)        ;;SAVE R0
8771  057262  005000                    CLR     R0              ;;PICKUP THE ITEM INDEX
8772  057264  153700  001114            BISB    @#$ITEMB,R0
8773  057270  001004                    BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
8774                                                            ;;TYPE THE PC OF THE ERROR
8775  057272  013746  001116            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
8776                                                            ;;ERROR ADDRESS
8777  057276  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8778  057300  000445                    BR      10$             ;;GET OUT
8779  057302  005300            1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
8780  057304  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
8781  057306  006300                    ASL     R0
8782  057310  006300                    ASL     R0
8783  057312  062700  001304            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
8784  057316  012037  057326            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
8785  057322  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
8786  057324  104401                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
8787  057326  000000            2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
8788  057330  104401  001207            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8789  057334  012037  057344    3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
8790  057340  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
8791  057342  104401                    TYPE                    ;;TYPE THE "DATA HEADER"
8792  057344  000000            4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
8793  057346  104401  001207            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
8794  057352  010146            5$:     MOV     R1,-(SP)        ;;SAVE R1
8795  057354  012001                    MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
8796  057356  001415                    BEQ     9$              ;;BR IF NO DATA TO BE TYPED
8797  057360  012000                    MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
8798  057362  105720            6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
8799  057364  001003                    BNE     7$              ;;BR IF DECIMAL
8800  057366  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
8801  057370  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8802  057372  000402                    BR      8$
8803  057374            7$:
8804  057374  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
```

```
 8805  057376  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8806  057400  005711            8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
 8807  057402  001403                    BEQ     9$              ;;BR IF NO
 8808  057404  104401  057424            TYPE    ,11$            ;;TYPE TWO(2) SPACES
 8809  057410  000764                    BR      6$              ;;LOOP
 8810
 8811  057412  012601            9$:     MOV     (SP)+,R1        ;;RESTORE R1
 8812  057414  012600            10$:    MOV     (SP)+,R0        ;;RESTORE R0
 8813  057416  104401  001207            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 8814  057422  000207                    RTS     PC              ;;RETURN
 8815  057424  020040      000   11$:     .ASCIZ  /  /            ;;TWO(2) SPACES
 8816          057430                    .EVEN
 8817                            .SBTTL  TYPE ROUTINE
 8818
 8819                            ;;**************************************************************
 8820                            ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 8821                            ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 8822                            ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 8823                            ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 8824                            ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 8825                            ;*
 8826                            ;*CALL:
 8827                            ;*1) USING A TRAP INSTRUCTION
 8828                            ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 8829                            ;*OR
 8830                            ;*      TYPE
 8831                            ;*      MESADR
 8832                            ;*
 8833
 8834  057430  105737  001157    $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 8835  057434  100002                    BPL     1$              ;;BR IF YES
 8836  057436  000000                    HALT                    ;;HALT HERE IF NO TERMINAL
 8837  057440  000407                    BR      3$              ;;LEAVE
 8838  057442  010046            1$:     MOV     R0,-(SP)        ;;SAVE R0
 8839  057444  017600  000002            MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 8840  057450  112046            2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 8841  057452  001005                    BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 8842  057454  005726                    TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 8843  057456  012600            60$:    MOV     (SP)+,R0        ;;RESTORE R0
 8844  057460  062716  000002    3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 8845  057464  000002                    RTI                     ;;RETURN
 8846  057466  122716  000011    4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 8847  057472  001430                    BEQ     8$
 8848  057474  122716  000200            CMPB    #CRLF (SP)      ;;BRANCH IF NOT <CRLF>
 8849  057500  001006                    BNE     5$
 8850  057502  005726                    TST     (SP)+           ;;POP  <CR><LF> EQUIV
 8851  057504  104401                    TYPE                    ;;TYPE A CR AND LF
 8852  057506  001207                    $CRLF
 8853  057510  105037  057644            CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 8854  057514  000755                    BR      2$              ;;GET NEXT CHARACTER
 8855  057516  004737  057600    5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 8856  057522  123726  001156    6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 8857  057526  001350                    BNE     2$              ;;IF NO GO GET NEXT CHAR.
 8858  057530  013746  001154            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 8859                                                            ;;AND THE NULL CHAR.
 885C  057534  105366  000001    7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
```

```
 8861  057540  002770                        BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
 8862  057542  004737  057600                JSR     PC,$TYPEC               ;;GO TYPE A NULL
 8863  057546  105337  057644                DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
 8864  057552  000770                        BR      7$                      ;;LOOP
 8865
 8866                                  ;HORIZONTAL TAB PROCESSOR
 8867
 8868  057554  112716  000040          8$:   MOVB    #' ,(SP)                ;;REPLACE TAB WITH SPACE
 8869  057560  004737  057600          9$:   JSR     PC,$TYPEC               ;;TYPE A SPACE
 8870  057564  132737  000007  057644        BITB    #7,$CHARCNT             ;;BRANCH IF NOT AT
 8871  057572  001372                        BNE     9$                      ;;TAB STOP
 8872  057574  005726                        TST     (SP)+                   ;;POP SPACE OFF STACK
 8873  057576  000724                        BR      2$                      ;;GET NEXT CHARACTER
 8874  057600  105777  121344          $TYPEC: TSTB  @$TPS                   ;;WAIT UNTIL PRINTER IS READY
 8875  057604  100375                        BPL     $TYPEC
 8876  057606  116677  000002  121336        MOVB    2(SP),@$TPB             ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 8877  057614  122766  000015  000002        CMPB    #CR,2(SP)               ;;IS CHARACTER A CARRIAGE RETURN?
 8878  057622  001003                        BNE     1$                      ;;BRANCH IF NO
 8879  057624  105037  057644                CLRB    $CHARCNT                ;;YES--CLEAR CHARACTER COUNT
 8880  057630  000406                        BR      $TYPEX                  ;;EXIT
 8881  057632  122766  000012  000002  1$:   CMPB    #LF,2(SP)               ;;IS CHARACTER A LINE FEED?
 8882  057640  001402                        BEQ     $TYPEX                  ;;BRANCH IF YES
 8883  057642  105227                        INCB    (PC)+                   ;;COUNT THE CHARACTER
 8884  057644  000000          $CHARCNT: .WORD 0                             ;;CHARACTER COUNT STORAGE
 8885  057646  000207          $TYPEX: RTS     PC
 8886
 8887                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 8888
 8889                                  ;;**************************************************************
 8890                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 8891                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 8892                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 8893                                  ;*CALL:
 8894                                  ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
 8895                                  ;*      TYPOS                           ;;CALL FOR TYPEOUT
 8896                                  ;*      .BYTE   N                       ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 8897                                  ;*      .BYTE   M                       ;;M=1 OR 0
 8898                                  ;*                                      ;;1=TYPE LEADING ZEROS
 8899                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 8900                                  ;*
 8901                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 8902                                  ;*$TYPOS OR $TYPOC
 8903                                  ;*CALL:
 8904                                  ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
 8905                                  ;*      TYPON                           ;;CALL FOR TYPEOUT
 8906                                  ;*
 8907                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 8908                                  ;*CALL:
 8909                                  ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
 8910                                  ;*      TYPOC                           ;;CALL FOR TYPEOUT
 8911
 8912  057650  017646  000000          $TYPOS: MOV     @(SP),-(SP)             ;;PICKUP THE MODE
 8913  057654  116637  000001  060073        MOVB    1(SP),$OFILL            ;;LOAD ZERO FILL SWITCH
 8914  057662  112637  060075                MOVB    (SP)+,$OMODE+1          ;;NUMBER OF DIGITS TO TYPE
 8915  057666  062716  000002                ADD     #2,(SP)                 ;;ADJUST RETURN ADDRESS
 8916  057672  000406                        BR      $TYPON
```

```
 8917  057674  112737  000001  060073  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 8918  057702  112737  000006  060075          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 8919  057710  112737  000005  060072  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 8920  057716  010346                          MOV     R3,-(SP)        ;;SAVE R3
 8921  057720  010446                          MOV     R4,-(SP)        ;;SAVE R4
 8922  057722  010546                          MOV     R5,-(SP)        ;;SAVE R5
 8923  057724  113704  060075                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 8924  057730  005404                          NEG     R4
 8925  057732  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 8926  057736  110437  060074                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 8927  057742  113704  060073                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 8928  057746  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 8929  057752  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 8930  057754  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 8931  057756  000404                          BR      3$              ;;GO DO MSB
 8932  057760  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 8933  057762  006105                          ROL     R5
 8934  057764  006105                          ROL     R5
 8935  057766  010503                          MOV     R5,R3
 8936  057770  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 8937  057772  105337  060074                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 8938  057776  100016                          BPL     7$              ;;BR IF NO
 8939  060000  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 8940  060004  001002                          BNE     4$              ;;TEST FOR 0
 8941  060006  005704                          TST     R4              ;;SUPPRESS THIS 0?
 8942  060010  001403                          BEQ     5$              ;;BR IF YES
 8943  060012  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 8944  060014  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 8945  060020  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 8946  060024  110337  060070                  MOVB    R3,8$           ;;SAVE FOR TYPING
 8947  060030  104401  060070                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 8948  060034  105337  060072          7$:     DECB    $OCNT           ;;COUNT BY 1
 8949  060040  003347                          BGT     2$              ;;BR IF MORE TO DO
 8950  060042  002402                          BLT     6$              ;;BR IF DONE
 8951  060044  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 8952  060046  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 8953  060050  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 8954  060052  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 8955  060054  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 8956  060056  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 8957  060064  012616                          MOV     (SP)+,(SP)
 8958  060066  000002                          RTI                     ;;RETURN
 8959  060070  000000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 8960  060071  000                             .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 8961  060072  000                     $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 8962  060073  000                     $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 8963  060074  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 8964                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 8965
 8966                                  ;;**************************************************************
 8967                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 8968                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 8969                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 8970                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 8971                                  ;*REPLACED WITH SPACES.
 8972                                  ;*CALL:
```

```
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                                        
                                        $TYPDS:
060076                                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
060076  010046                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
060100  010146                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
060102  010246                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
060104  010346                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
060106  010546                                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
060110  012746  020200                        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
060114  016605  000020                        BPL     1$              ;;BR IF INPUT IS POS.
060120  100004                                 NEG     R5              ;;MAKE THE BINARY NUMBER POS.
060122  005405                                 MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
060124  112766  000055  000001                 CLR     R0              ;;ZERO THE CONSTANTS INDEX
060132  005000                        1$:     MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
060134  012703  060312                        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
060140  112723  000040                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
060144  005002                        3$:     MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
060146  016001  060302                        SUB     R1,R5           ;;FORM THIS BCD DIGIT
060152  160105                                 BLT     4$              ;;BR IF DONE
060154  002402                                 INC     R2              ;;INCREASE THE BCD DIGIT BY 1
060156  005202                                 BR      3$
060160  000774                        4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
060162  060105                                 TST     R2              ;;CHECK IF BCD DIGIT=0
060164  005702                                 BNE     5$              ;;FALL THROUGH IF 0
060166  001002                                 TSTB    (SP)            ;;STILL DOING LEADING 0'S?
060170  105716                                 BMI     7$              ;;BR IF YES
060172  100407                        5$:     ASLB    (SP)            ;;MSD?
060174  106316                                 BCC     6$              ;;BR IF NO
060176  103003                                 MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
060200  116663  000001  177777                 BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
060206  052702  000060                 6$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
060212  052702  000040                 7$:     MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
060216  110223                                 TST     (R0)+           ;;JUST INCREMENTING
060220  005720                                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
060222  020027  000010                        BLT     2$              ;;GO DO THE NEXT DIGIT
060226  002746                                 BGT     8$              ;;GO TO EXIT
060230  003002                                 MOV     R5,R2           ;;GET THE LSD
060232  010502                                 BR      6$              ;;GO CHANGE TO ASCII
060234  000764                        8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
060236  105726                                 BPL     9$              ;;BR IF NO
060240  100003                                 MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
060242  116663  177777  177776                 CLRB    (R3)            ;;SET THE TERMINATOR
060250  105013                        9$:     MOV     (SP)+,R5        ;;POP STACK INTO R5
060252  012605                                 MOV     (SP)+,R3        ;;POP STACK INTO R3
060254  012603                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
060256  012602                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
060260  012601                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
060262  012600                                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
060264  104401  060312                        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
060270  016666  000002  000004                 MOV     (SP)+,(SP)
060276  012616                                 RTI                     ;;RETURN TO USER
060300  000002
060302  023420                        $DTBL:  10000.
060304  001750                                1000.
060306  000144                                100.
```

```
                                        10.
060310  000012                  $DBLK:  .BLKW   4
060312  000004                  .SBTTL  TTY INPUT ROUTINE
                                ;
                                ;;*****************************************************************
                                .ENABL  LSB
060322  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
060324  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
060326  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
060330  000001                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        060331                  $TKQEND=.
        060332                  .EVEN
                                ;
                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
060332  005037  060322          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
060336  012737  060330  060324          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
060344  013737  060324  060326          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
060352  012737  060402  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
060360  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
060366  005777  120554                  TST     @$TKB           ;;CLEAR DONE FLAG
060372  012777  000100  120544          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
060400  000207                        RTS     PC              ;;RETURN TO CALLER
                                ;
                                ;*TK SERVICE ROUTINE
                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                ;*IT IN THE QUEUE.
                                ;
060402  117746  120540          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
060406  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
060412  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
060416  001004                          BNE     2$              ;;BRANCH IF NO
060420  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
060426  001500                          BEQ     6$              ;;GO TO SWR CHANGE
                                ;
060430                          2$:
060430  022737  000001  060322          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
060436  001004                          BNE     3$              ;;BRANCH IF NO
060440  104401  001202                  TYPE    ,$BELL          ;;RING THE TTY BELL
060444  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
060446  000451                          BR      5$              ;;EXIT
060450  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
060454  001021                          BNE     32$             ;;BRANCH IF NO
060456  005077  120462                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
060462  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
060464  105777  120454          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
060470  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
060472  117746  120450                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
060476  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
```

```
 9085  060502  022627  000021              CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 9086  060506  001366                      BNE     31$             ;;BRANCH IF NO
 9087  060510  012777  000100  120426      MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 9088  060516  000002                      RTI                     ;;RETURN
 9089  060520  005237  060322      32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 9090  060524  021627  000140              CMP     (SP),#140       ;;IS IT UPPER CASE?
 9091  060530  002405                      BLT     4$              ;;BRANCH IF YES
 9092  060532  021627  000175              CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 9093  060536  003002                      BGT     4$              ;;BRANCH IF YES
 9094  060540  042716  000040              BIC     #40,(SP)        ;;MAKE IT UPPER CASE
 9095  060544  112677  177554      4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 9096  060550  005237  060324              INC     $TKQIN          ;;UPDATE THE POINTER
 9097  060554  023727  060324  060331      CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 9098  060562  001003                      BNE     5$              ;;BRANCH IF NO
 9099  060564  012737  060330  060324      MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 9100  060572  000002              5$:     RTI                     ;;RETURN
 9101
 9102                                      ;;**************************************************************
 9103                                      ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 9104                                      ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
 9105                                      ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
 9106                                      ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 9107  060574  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
 9108  060602  001104                      BNE     15$             ;;EXIT IF NOT
 9109  060604  105777  120334              TSTB    @$TKS           ;;IS A CHAR WAITING?
 9110  060610  100101                      BPL     15$             ;;IF NOT, EXIT
 9111  060612  117746  120330              MOVB    @$TKB,-(SP)     ;;YES
 9112  060616  042716  177600              BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
 9113  060622  021627  000007              CMP     (SP),#7         ;;IS IT A CONTROL-G?
 9114  060626  001300                      BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 9115                                                              ;;AND EXIT
 9116
 9117                                      ;;**************************************************************
 9118                                      ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 9119                                      ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 9120                                      ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 9121  060630  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 9122  060636  001674                      BEQ     2$              ;;BRANCH IF YES
 9123  060640  005726                      TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 9124  060642  004737  060332              JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 9125  060646  005077  120272              CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9126  060652  112737  000001  001135      MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
 9127
 9128  060660  104401  061436              TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 9129  060664  104401  061443      $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 9130  060670  013746  000176              MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 9131  060674  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS
 9132  060676  104401  061454              TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 9133  060702  005046              19$:    CLR     -(SP)           ;;CLEAR COUNTER
 9134  060704  005046                      CLR     -(SP)           ;;THE NEW SWR
 9135  060706  105777  120232      7$:     TSTB    @$TKS           ;;CHAR THERE?
 9136  060712  100375                      BPL     7$              ;;IF NOT TRY AGAIN
 9137
 9138  060714  117746  120226              MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 9139  060720  042716  177600              BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9140
```

```
9141
9142
9143  060724  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
9144  060730  001005                          BNE     10$             ;;BRANCH IF NOT
9145  060732  104401  061431                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
9146  060736  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
9147  060742  000757                          BR      19$             ;;LET'S TRY IT AGAIN
9148
9149
9150  060744  021627  000015          13$:    CMP     (SP),#15        ;;IS IT A <CR>?
9151  060750  001022                          BNE     16$             ;;BRANCH IF NO
9152  060752  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
9153  060756  001403                          BEQ     11$             ;;BRANCH IF YES
9154  060760  016677  000002  120152          MOV     2(SP),@SWR      ;;SAVE NEW SWR
9155  060766  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
9156  060772  104401  001207          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
9157  060776  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
9158  061004  001003                          BNE     15$             ;;BRANCH IF NOT
9159  061006  012777  000100  120130          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
9160  061014  000002          15$:    RTI                             ;;RETURN
9161  061016  004737  057600          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
9162  061022  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
9163  061026  002420                          BLT     18$             ;;BRANCH IF YES
9164  061030  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
9165  061034  003015                          BGT     18$             ;;BRANCH IF YES
9166  061036  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
9167  061042  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
9168  061046  001403                          BEQ     17$             ;;BRANCH IF YES
9169  061050  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
9170  061052  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
9171  061054  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
9172  061056  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
9173  061062  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
9174  061066  000707                          BR      7$              ;;GET THE NEXT ONE
9175  061070  104401  001206          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
9176  061074  000720                          BR      20$             ;;SIMULATE CONTROL-U
9177                                  .DSABL  LSB
9178
9179
9180                                  ;;*****************************************************************
9181                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9182                                  ;*CALL:
9183                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
9184                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
9185                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
9186                                  ;
9187
9188  061076  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
9189  061100  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
9190  061106  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
9191  061112  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
9192  061114  012746  061122                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
9193  061120  000002                          RTI                     ;;POP NEW PC AND PS
9194  061122                          64$:
9195  061122  005737  060322          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
9196  061126  001775                          BEQ     1$
```

```
 9197  061130  005337  060322                 DEC     $TKCNT                  ;;DECREMENT THE COUNTER
 9198  061134  117766  177166  000004         MOVB    @$TKQOUT,4(SP)          ;;GET ONE CHARACTER
 9199  061142  005237  060326                 INC     $TKQOUT                 ;;UPDATE THE POINTER
 9200  061146  023727  060326  060331         CMP     $TKQOUT,#$TKQEND        ;;DID IT GO OFF OF THE END?
 9201  061154  001003                         BNE     2$                      ;;BRANCH IF NO
 9202  061156  012737  060330  060326         MOV     #$TKQSRT,$TKQOUT        ;;RESET THE POINTER
 9203  061164  000002                 2$:     RTI                             ;;RETURN
 9204                                  ;;*****************************************************************
 9205                                  ;;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 9206                                  ;;*CALL:
 9207                                  ;;*     RDLIN                           ;;INPUT A STRING FROM THE TTY
 9208                                  ;;*     RETURN HERE                     ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 9209                                  ;;*                                     ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 9210
 9211  061166  010346                 $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
 9212  061170  005046                         CLR     -(SP)                   ;;CLEAR THE RUBOUT KEY
 9213  061172  012703  061422         1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
 9214  061176  022703  061431         2$:     CMP     #$TTYIN+7.,R3           ;;BUFFER FULL?
 9215  061202  101456                         BLOS    4$                      ;;BR IF YES
 9216  061204  104410                         RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
 9217  061206  112613                         MOVB    (SP)+,(R3)              ;;GET CHARACTER
 9218  061210  122713  000177         10$:    CMPB    #177,(R3)               ;;IS IT A RUBOUT
 9219  061214  001022                         BNE     5$                      ;;BR IF NO
 9220  061216  005716                         TST     (SP)                    ;;IS THIS THE FIRST RUBOUT?
 9221  061220  001007                         BNE     6$                      ;;BR IF NO
 9222  061222  112737  000134  061420         MOVB    #'\,9$                  ;;TYPE A BACK SLASH
 9223  061230  104401  061420                 TYPE    ,9$
 9224  061234  012716  177777                 MOV     #-1,(SP)                ;;SET THE RUBOUT KEY
 9225  061240  005303                 6$:     DEC     R3                      ;;BACKUP BY ONE
 9226  061242  020327  061422                 CMP     R3,#$TTYIN              ;;STACK EMPTY?
 9227  061246  103434                         BLO     4$                      ;;BR IF YES
 9228  061250  111337  061420                 MOVB    (R3),9$                 ;;SETUP TO TYPEOUT THE DELETED CHAR.
 9229  061254  104401  061420                 TYPE    ,9$                     ;;GO TYPE
 9230  061260  000746                         BR      2$                      ;;GO READ ANOTHER CHAR.
 9231  061262  005716                 5$:     TST     (SP)                    ;;RUBOUT KEY SET?
 9232  061264  001406                         BEQ     7$                      ;;BR IF NO
 9233  061266  112737  000134  061420         MOVB    #'\,9$                  ;;TYPE A BACK SLASH
 9234  061274  104401  061420                 TYPE    ,9$
 9235  061300  005016                         CLR     (SP)                    ;;CLEAR THE RUBOUT KEY
 9236  061302  122713  000025         7$:     CMPB    #25,(R3)                ;;IS CHARACTER A CTRL U?
 9237  061306  001003                         BNE     8$                      ;;BR IF NO
 9238  061310  104401  061431                 TYPE    ,$CNTLU                 ;;TYPE A CONTROL "U"
 9239  061314  000726                         BR      1$                      ;;GO START OVER
 9240  061316  122713  000022         8$:     CMPB    #22,(R3)                ;;IS CHARACTER A "^R"?
 9241  061322  001011                         BNE     3$                      ;;BRANCH IF NO
 9242  061324  105013                         CLRB    (R3)                    ;;CLEAR THE CHARACTER
 9243  061326  104401  001207                 TYPE    ,$CRLF                  ;;TYPE A "CR" & "LF"
 9244  061332  104401  061422                 TYPE    ,$TTYIN                 ;;TYPE THE INPUT STRING
 9245  061336  000717                         BR      2$                      ;;GO PICKUP ANOTHER CHACTER
 9246  061340  104401  001206         4$:     TYPE    ,$QUES                  ;;TYPE A '?'
 9247  061344  000712                         BR      1$                      ;;CLEAR THE BUFFER AND LOOP
 9248  061346  111337  061420         3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
 9249  061352  104401  061420                 TYPE    ,9$
 9250  061356  122723  000015                 CMPB    #15,(R3)+               ;;CHECK FOR RETURN
 9251  061362  001305                         BNE     2$                      ;;LOOP IF NOT RETURN
 9252  061364  105063  177777                 CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
```

```
 9253  061370  104401  001210                  TYPE    $LF             ;;TYPE A LINE FEED
 9254  061374  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 9255  061376  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 9256  061400  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 9257  061402  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 9258  061410  012766  061422  000004          MOV     #$TTYIN,4(SP)
 9259  061416  000002                          RTI                     ;;RETURN
 9260  061420     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 9261  061421     000                          .BYTE   0               ;;TERMINATOR
 9262  061422  000007                  $TTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
 9263  061431     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 9264  061436  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 9265  061443     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
 9266  061450  036440  000040
 9267  061454  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
 9268  061462  020075     000
 9269          061466                  .EVEN
 9270                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 9271
 9272                                  ;;**************************************************************
 9273                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 9274                                  ;*CHANGE IT TO BINARY.
 9275                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
 9276                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
 9277                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
 9278                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
 9279                                  ;*CALL:
 9280                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 9291                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 9282                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 9283
 9284  061466  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 9285  061470  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
 9286  061476  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9287  061500  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9288  061502  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9289  061504  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
 9290  061506  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 9291  061510  010037  061614                  MOV     R0,5$           ;;AND SAVE IT
 9292  061514  005001                          CLR     R1              ;;CLEAR DATA WORD
 9293  061516  005002                          CLR     R2
 9294  061520  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 9295  061522  001420                          BEQ     3$              ;;IF ZERO GET OUT
 9296  061524  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
 9297  061530  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
 9298  061532  122716  000067                  CMPB    #'7,(SP)
 9299  061536  002423                          BLT     4$
 9300  061540  006301                          ASL     R1              ;;*2
 9301  061542  006102                          ROL     R2
 9302  061544  006301                          ASL     R1              ;;*4
 9303  061546  006102                          ROL     R2
 9304  061550  006301                          ASL     R1              ;;*8
 9305  061552  006102                          ROL     R2
 9306  061554  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 9307  061560  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 9308  061562  000756                          BR      2$              ;;LOOP
```

```
9309  061564  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
9310  061566  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
9311  061572  012700  061624                  MOV     #$IOCT,R0
9312  061576  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
9313  061600  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9314  061602  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9315  061604  000002                          RTI                     ;;RETURN
9316  061606  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
9317  061610  105010                          CLRB    (R0)            ;;SET A TERMINATOR
9318  061612  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
9319  061614  000000                  5$:     .WORD   0
9320  061616  104401  001206                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
9321  061622  000730                          BR      1$              ;;TRY AGAIN
9322  061624  000000                  $IOCT:  .WORD   0               ;;HIGH ORDER BITS GO HERE
9323                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
9324
9325                                  ;;******************************************************************
9326                                  ;*SAVE R0-R5
9327                                  ;*CALL:
9328                                  ;*      SAVREG
9329                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
9330                                  ;*
9331                                  ;*TOP---(+16)
9332                                  ;* +2---(+18)
9333                                  ;* +4---R5
9334                                  ;* +6---R4
9335                                  ;* +8---R3
9336                                  ;*+10---R2
9337                                  ;*+12---R1
9338                                  ;*+14---R0
9339
9340  061626                          $SAVREG:
9341  061626  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9342  061630  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9343  061632  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9344  061634  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
9345  061636  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
9346  061640  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9347  061642  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
9348  061646  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
9349  061652  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
9350  061656  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
9351  061662  000002                          RTI
9352
9353                                  ;*RESTORE R0-R5
9354                                  ;*CALL:
9355                                  ;*      RESREG
9356  061664                          $RESREG:
9357  061664  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
9358  061670  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
9359  061674  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
9360  061700  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
9361  061704  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
9362  061706  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
9363  061710  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
9364  061712  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        RTI
                                .SBTTL  TRAP DECODER

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

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  055012  026532  TITLE:  .ASCII  <CR><LF><LF>C2-C2RJE-B<CR><LF>
```

```
9421   062016  055103  045122  026505
9422   062024  006502     012
9423   062027     122  030120  027464          .ASCIZ  @RP04/5/6 DUAL CONTROLLER LOGIC TEST - PART 1@<CR><LF><LF>
9424   062034  027465  020066  052504
9425   062042  046101  041440  047117
9426   062050  051124  046117  042514
9427   062056  020122  047514  044507
9428   062064  020103  042524  052123
9429   062072  026440  050040  051101
9430   062100  020124  006461  005012
9431   062106     000
9432   062107     015  042412  052116  ENTERA: .ASCIZ  <CR><LF>/ENTER DRIVE ADDRESS: /
9433   062114  051105  042040  044522
9434   062122  042526  040440  042104
9435   062130  042522  051523  020072
9436   062136     000
9437   062137     111  053116  046101  ADRERR: .ASCIZ  /INVALID ADDRESS/<CR><LF>
9438   062144  042111  040440  042104
9439   062152  042522  051523  005015
9440   062160     000
9441   062161     015  050012  051117  PORTAIS: .ASCIZ <CR><LF>/PORT A ADDRESS IS: /
9442   062166  020124  020101  042101
9443   062174  051104  051505  020123
9444   062202  051511  020072     000
9445   062207     015  050012  051117  PORTBIS: .ASCIZ <CR><LF>/PORT B ADDRESS IS: /
9446   062214  020124  020102  042101
9447   062222  051104  051505  020123
9448   062230  051511  020072     000
9449   062235     015  051412  051531  NOCLOCK: .ASCIZ <CR><LF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CR><LF><LF
9450   062242  042524  020115  052515
9451   062250  052123  044040  053101
9452   062256  020105  046047  020047
9453   062264  051117  023440  023520
9454   062272  041440  047514  045503
9455   062300  005015  000012
9456   062304  042412  052116  051105  TESTNO: .ASCIZ  <LF>/ENTER TEST #: /
9457   062312  052040  051505  020124
9458   062320  035043  000040
9459   062324  047111  040526  044514  BADNO:  .ASCIZ  /INVALID TEST NUMBER/<CR><LF>
9460   062332  020104  042524  052123
9461   062340  047040  046525  042502
9462   062346  006522  000012
9463   062352  042440  051122  051117  TSTERR: .ASCIZ  / ERRORS/<CR><LF>
9464   062360  006523  000012
9465   062364  005015  052012  042510  ADDRIS: .ASCIZ  <CR><LF><LF>/THE PRESENT ADDRESS OF THE RH11 (RPCS1) IS:
9466   062372  050040  042522  042523
9467   062400  052116  040440  042104
9468   062406  042522  051523  047440
9469   062414  020106  044124  020105
9470   062422  044122  030461  024040
9471   062430  050122  051503  024461
9472   062436  044440  035123  000040
9473   062444  042412  052116  051105  NTRH11: .ASCIZ  <LF>/ENTER NEW RH11 ADDRESS: /
9474   062452  047040  053505  051040
9475   062460  030510  020061  042101
9476   062466  051104  051505  035123
```

```
9477  062474  000040
9478
9479                          ;;*****************************************************************
9480
9481                          .SBTTL  TEST ERROR MESSAGES
9482
9483                          ;;*****************************************************************
9484
9485  062476  051127  047117  020107  EM1:    .ASCIZ  /WRONG DRIVE TYPE/
9486  062504  051104  053111  020105
9487  062512  054524  042520     000
9488
9489  062517     104  044522  042526  EM2:    .ASCIZ  /DRIVE NOT ON LINE/
9490  062524  047040  052117  047440
9491  062532  020116  044514  042516
9492  062540     000
9493
9494  062541     123  051105  040511  EM3:    .ASCIZ  /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME
9495  062546  020114  052516  041115
9496  062554  051105  051040  040505
9497  062562  020104  044124  047522
9498  062570  043525  020110  040505
9499  062576  044103  050040  051117
9500  062604  020124  047516  020124
9501  062612  044124  020105  040523
9502  062620  042515     000
9503
9504  062623     104  044522  042526  EM4:    .ASCIZ  /DRIVE NOT SEIZED BY PORT/
9505  062630  047040  052117  051440
9506  062636  044505  042532  020104
9507  062644  054502  050040  051117
9508  062652  000124
9509
9510  062654  051127  047117  020107  EM5:    .ASCIZ  /WRONG STATUS SEEN BY THE SEIZING PORT/
9511  062662  052123  052101  051525
9512  062670  051440  042505  020116
9513  062676  054502  052040  042510
9514  062704  051440  044505  044532
9515  062712  043516  050040  051117
9516  062720  000124
9517
9518  062722  042522  044507  052123  EM6:    .ASCIZ  /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED
9519  062730  051105  041440  047117
9520  062736  042524  052116  020123
9521  062744  042527  042522  051440
9522  062752  042505  020116  054502
9523  062760  047440  050120  051517
9524  062766  052111  020105  047520
9525  062774  052122  026440  042040
9526  063002  044522  042526  053440
9527  063010  051501  051440  044505
9528  063016  042532  000104
9529
9530  063022  042522  044507  052123  EM7:    .ASCIZ  REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT
9531  063030  051105  041440  047117
9532  063036  042524  052116  020123
```

```
 9533  063044  051127  047117  020107
 9534  063052  043101  042524  020122
 9535  063060  042522  042514  051501
 9536  063066  020105  051117  052040
 9537  063074  046511  047505  052125
 9538  063102     000
 9539
 9540  063103     122  043505  051511  EM10:   .ASCIZ  /REGISTER CONTENTS WRONG/
 9541  063110  042524  020122  047503
 9542  063116  052116  047105  051524
 9543  063124  053440  047522  043516
 9544  063132     000
 9545
 9546  063133     103  047117  051124  EM11:   .ASCIZ  /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
 9547  063140  046117  041040  051525
 9548  063146  050040  051101  052111
 9549  063154  020131  051105  047522
 9550  063162  020122  042522  042101
 9551  063170  047111  020107  047111
 9552  063176  044504  040503  042524
 9553  063204  020104  042522  044507
 9554  063212  052123  051105     000
 9555
 9556  063217     104  044522  042526  EM12:   .ASCIZ  /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
 9557  063224  047040  052117  051440
 9558  063232  044505  042532  020104
 9559  063240  054502  042040  044522
 9560  063246  042526  041440  042514
 9561  063254  051101  041440  046517
 9562  063262  040515  042116     000
 9563
 9564  063267     122  040505  044504  EM13:   .ASCIZ  /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT
 9565  063274  020116  051120  051505
 9566  063302  052105  042040  042517
 9567  063310  020123  047516  020124
 9568  063316  042523  020124  047526
 9569  063324  052514  042515  053040
 9570  063332  046101  042111  043040
 9571  063340  051117  052040  042510
 9572  063346  050040  051117  000124
 9573
 9574  063354  047526  052514  042515  EM14:   .ASCIZ  /VOLUME VALID SET ON THE WRONG PORT
 9575  063362  053040  046101  042111
 9576  063370  051440  052105  047440
 9577  063376  020116  044124  020105
 9578  063404  051127  047117  020107
 9579  063412  047520  052122     000
 9580
 9581  063417     101  052124  020116  EM15:   .ASCIZ  /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
 9582  063424  044502  020124  051127
 9583  063432  047117  020107  043101
 9584  063440  042524  020122  044524
 9585  063446  042515  052517  020124
 9586  063454  020055  042522  052521
 9587  063462  051505  020124  047516
 9588  063470  020124  042523  000124
```

```
 9589
 9590  063476  052101  047124  041040  EM16:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
 9591  063504  052111  053440  047522
 9592  063512  043516  040440  052106
 9593  063520  051105  051040  046105
 9594  063526  040505  042523  026440
 9595  063534  051040  050505  042525
 9596  063542  052123  051440  052105
 9597  063550     000
 9598
 9599  063551     101  052124  020116  EM17:   .ASCIZ  /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
 9600  063556  044502  020124  051127
 9601  063564  047117  020107  043101
 9602  063572  042524  020122  042522
 9603  063600  042514  051501  020105
 9604  063606  020055  042522  052521
 9605  063614  051505  020124  047516
 9606  063622  020124  042523  000124
 9607
 9608  063630  051104  053111  020105  EM20:   .ASCIZ  /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
 9609  063636  047516  020124  042523
 9610  063644  055111  042105  053440
 9611  063652  042510  020116  052101
 9612  063660  047124  041040  052111
 9613  063666  043040  051117  050040
 9614  063674  051117  020124  046103
 9615  063702  040505  042522  000104
 9616
 9617  063710  051104  053111  020105  EM21:   .ASCIZ  /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
 9618  063716  042523  055111  042105
 9619  063724  053440  042510  020116
 9620  063732  042532  047522  053440
 9621  063740  044522  052124  047105
 9622  063746  044440  020116  052101
 9623  063754  047124  041040  052111
 9624  063762     000
 9625
 9626  063763     104  044522  042526  EM22:   .ASCIZ  /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET
 9627  063770  047040  052117  044440
 9628  063776  020116  042516  052125
 9629  064004  040522  020114  043101
 9630  064012  042524  020122  044524
 9631  064020  042515  052517  020124
 9632  064026  020055  042522  052521
 9633  064034  051505  020124  047516
 9634  064042  020124  042523  000124
 9635
 9636  064050  044524  042515  052517  EM23:   .ASCIZ  /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
 9637  064056  020124  046103  040505
 9638  064064  042522  020104  044124
 9639  064072  020105  051104  053111
 9640  064100  023505  020123  051105
 9641  064106  047522  020122  044502
 9642  064114  000124
 9643
 9644  064116  042522  042514  051501  EM24:   .ASCIZ   RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
```

```
 9645  064124  020105  047503  046515
 9646  064132  047101  020104  042522
 9647  064140  042514  051501  042105
 9648  064146  042040  044522  042526
 9649  064154  053440  052111  020110
 9650  064162  051105  047522  051522
 9651  064170  051440  052105     000
 9652
 9653  064175     124  046511  047505  EM25:   .ASCIZ   TIMEOUT ONE-SHOT DID NOT RETRIGGER/
 9654  064202  052125  047440  042516
 9655  064210  051455  047510  020124
 9656  064216  044504  020104  047516
 9657  064224  020124  042522  051124
 9658  064232  043511  042507  000122
 9659
 9660  064240  051104  053111  020105  EM26:   .ASCIZ   DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
 9661  064246  047516  020124  047111
 9662  064254  047040  052505  051124
 9663  064262  046101  040440  052106
 9664  064270  051105  051040  046105
 9665  064276  040505  042523  026440
 9666  064304  051040  050505  042525
 9667  064312  052123  047040  052117
 9668  064320  051440  052105     000
 9669
 9670  064325     122  043505  051511  EM27:   .ASCIZ  /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
 9671  064332  042524  020122  051127
 9672  064340  047117  020107  043101
 9673  064346  042524  020122  042522
 9674  064354  042514  051501  020105
 9675  064362  044527  044124  051040
 9676  064370  050505  042525  052123
 9677  064376  051440  052105     000
 9678
 9679  064403     104  044522  042526  EM30:   .ASCIZ  /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
 9680  064410  051440  044505  042532
 9681  064416  020104  054502  051040
 9682  064424  046105  040505  042523
 9683  064432  041440  046517  040515
 9684  064440  042116  044440  051523
 9685  064446  042525  020104  044127
 9686  064454  047105  042040  044522
 9687  064462  042526  044440  020116
 9688  064470  042516  052125  040522
 9689  064476  000114
 9690
 9691  064500  051104  053111  020105  EM31:   .ASCIZ  /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
 9692  064506  047111  047040  052505
 9693  064514  051124  046101  040440
 9694  064522  052106  051105  051040
 9695  064530  046105  040505  042523
 9696  064536  026440  051040  050505
 9697  064544  042525  052123  051440
 9698  064552  052105     000
 9699
 9700  064555     101  052124  020116  EM32:   .ASCIZ   ATTN BIT WRONG AFTER RECALIBRATE COMMAND
```

```
9701   064562  044502  020124  051127
9702   064570  047117  020107  043101
9703   064576  042524  020122  042522
9704   064604  040503  044514  051102
9705   064612  052101  020105  047503
9706   064620  046515  047101  000104
9707
9708   064626  051104  053111  020105  EM33:   .ASCIZ  /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
9709   064634  042522  052524  047122
9710   064642  042105  052040  020117
9711   064650  042516  052125  040522
9712   064656  020114  043111  042040
9713   064664  044522  042526  041440
9714   064672  042514  051101  043440
9715   064700  053111  047105  053440
9716   064706  044510  042514  042040
9717   064714  044522  042526  051440
9718   064722  044505  042532  000104
9719
9720   064730  051104  053111  020105  EM34:   .ASCIZ   DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
9721   064736  042522  052524  047122
9722   064744  042105  052040  020117
9723   064752  042516  052125  040522
9724   064760  020114  043111  046440
9725   064766  051501  041123  051525
9726   064774  044440  044516  020124
9727   065002  044507  042526  020116
9728   065010  044127  046111  020105
9729   065016  051104  053111  020105
9730   065024  042523  055111  042105
9731   065032     000
9732
```

```
 9733  065033     124  046511  047505  EM35:   .ASCIZ   TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
 9734  065040  052125  047440  042516
 9735  065046  051440  047510  020124
 9736  065054  044506  042522  020104
 9737  065062  044527  044124  052517
 9738  065070  020124  042522  044507
 9739  065076  052123  051105  040440
 9740  065104  041503  051505  000123
 9741
 9742  065112  044524  042515  052517  EM36:   .ASCIZ  /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
 9743  065120  020124  040510  020123
 9744  065126  047516  020124  041517
 9745  065134  052503  051122  042105
 9746  065142  053440  052111  044510
 9747  065150  020116  020062  042523
 9748  065156  047503  042116  000123
 9749
 9750  065164  051104  053111  020105  EM37:   .ASCIZ  /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
 9751  065172  051511  047040  047117
 9752  065200  042455  044530  052123
 9753  065206  047105  020124  023450
 9754  065214  042516  023504  041040
 9755  065222  052111  051440  052105
 9756  065230  000051
 9757
 9758  065232  052101  047124  041040  EM40:   .ASCIZ  ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
 9759  065240  052111  043040  051117
 9760  065246  050040  051117  020124
 9761  065254  047516  020124  042522
 9762  065262  042523  020124  054502
 9763  065270  046440  051501  041123
 9764  065276  051525  041440  042514
 9765  065304  051101     000
 9766
 9767  065307     124  046511  047505  EM41:   .ASCIZ  /TIMEOUT CLEARED THE ATTENTION BIT/
 9768  065314  052125  041440  042514
 9769  065322  051101  042105  052040
 9770  065330  042510  040440  052124
 9771  065336  047105  044524  047117
 9772  065344  041040  052111     000
 9773
 9774  065351     104  044522  042526  EM42:   .ASCIZ  /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
 9775  065356  047040  052117  044440
 9776  065364  020116  042516  052125
 9777  065372  040522  020114  051117
 9778  065400  051440  044505  042532
 9779  065406  020104  043101  042524
 9780  065414  020122  052101  047124
 9781  065422  041040  052111  053440
 9782  065430  044522  052124  047105
 9783  065436     000
 9784
 9785  065437     104  044522  042526  EM43:   .ASCIZ  /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
 9786  065444  044440  020116  042516
 9787  065452  052125  040522  020114
 9788  065460  043101  042524  020122
```

```
9789  065466  052101  042524  052116
9790  065474  047511  020116  044502
9791  065502  020124  051127  052111
9792  065510  042524  000116
9793
9794  065514  051127  052111  020105  EM44:   .ASCIZ   WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
9795  065522  052101  042524  052116
9796  065530  047511  020116  044502
9797  065536  020124  044504  020104
9798  065544  047516  020124  042523
9799  065552  020124  047520  052122
9800  065560  051040  050505  042525
9801  065566  052123     000
9802
9803  065571     103  047117  051124  EM45:   .ASCIZ  @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A-B'@
9804  065576  046117  042514  020122
9805  065604  042523  042514  052103
9806  065612  051440  044527  041524
9807  065620  020110  047117  042040
9808  065626  044522  042526  047040
9809  065634  052117  044440  020116
9810  065642  040447  041057  000047
9811
9812  065650  040503  023516  020124  EM46:   .ASCIZ  /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
9813  065656  041501  042503  051523
9814  065664  042040  044522  042526
9815  065672  052040  051110  052517
9816  065700  044107  042440  052111
9817  065706  042510  020122  047520
9818  065714  052122     000
9819
9820  065717     101  052124  020116  EM47:   .ASCIZ  /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
9821  065724  044502  020124  047506
9822  065732  020122  042523  055111
9823  065740  047111  020107  047520
9824  065746  052122  047040  052117
9825  065754  041440  042514  051101
9826  065762  042105  041040  020131
9827  065770  040515  051523  052502
9828  065776  020123  047111  052111
9829  066004     000
9830
9831  066005     101  052124  020116  EM50:   .ASCIZ  /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
9832  066012  044502  020124  047506
9833  066020  020122  050117  047520
9834  066026  044523  042524  050040
9835  066034  051117  020124  046103
9836  066042  040505  042522  020104
9837  066050  054502  046440  051501
9838  066056  041123  051525  044440
9839  066064  044516  000124
9840
9841  066070  052101  047124  041040  EM51:   .ASCIZ  /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
9842  066076  052111  041440  042514
9843  066104  051101  042105  041040
9844  066112  020131  040515  051523
```

```
 9845  066120  052502  020123  047111
 9846  066126  052111  020054  051104
 9847  066134  053111  020105  047111
 9848  066142  047040  052505  051124
 9849  066150  046101     000
 9850
 9851  066153     124  042510  040440  EM52:   .ASCIZ  /THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
 9852  066160  052124  020116  044502
 9853  066166  020124  051511  051440
 9854  066174  052105  040440  052106
 9855  066202  051105  052040  046511
 9856  066210  047505  052125  053440
 9857  066216  052111  020110  047516
 9858  066224  051040  050505  042525
 9859  066232  052123  023040  023440
 9860  066240  051105  023522  051440
 9861  066246  052105     000
 9862
 9863  066251     103  047101  052047  EM53:   .ASCIZ  /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
 9864  066256  051040  040505  020104
 9865  066264  044124  020105  052101
 9866  066272  047124  041040  052111
 9867  066300  043040  047522  020115
 9868  066306  044124  020105  047447
 9869  066314  050120  051517  052111
 9870  066322  023505  050040  051117
 9871  066330  000124
 9872
 9873  066332  042522  042514  051501  EM54:   .ASCIZ  /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
 9874  066340  020105  047503  046515
 9875  066346  047101  020104  042522
 9876  066354  047503  047107  055111
 9877  066362  042105  053440  042510
 9878  066370  020116  051511  052523
 9879  066376  042105  041040  020131
 9880  066404  047516  026516  042523
 9881  066412  055111  047111  020107
 9882  066420  047520  052122     000
 9883
 9884  066425     124  046511  047505  EM55:    ASCIZ   TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
 9885  066432  052125  047440  042516
 9886  066440  051455  047510  020124
 9887  066446  051511  046040  051505
 9888  066454  020123  044124  047101
 9889  066462  032440  030060  046440
 9890  066470  000123
 9891
 9892  066472  044122  030461  042040  EM56:   .ASCIZ   RH11 DIDN'T RESPOND TO ADDRESSING
 9893  066500  042111  023516  020124
 9894  066506  042522  050123  047117
 9895  066514  020104  047524  040440
 9896  066522  042104  042522  051523
 9897  066530  047111  000107
 9898
 9899
 9900
```

M14


```
9901
9902
9903  066534 042524 052123 021440  DH1:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
9904  066542 020040 051105 020122
9905  066550 041520 020040 047520
9906  066556 052122 021440 020040
9907  066564 042522 020107 042101
9908  066572 020122 047503 052116
9909  066600 047105 051524    000
9910  066605    124 051505 020124  DH3:    .ASCIZ  /TEST #  ERR PC  REG ADR PORT A  PORT B/
9911  066612 020043 042440 051122
9912  066620 050040 020103 051040
9913  066626 043505 040440 051104
9914  066634 050040 051117 020124
9915  066642 020101 050040 051117
9916  066650 020124 000102
9917  066654 020040 020040 020040  DH4:    .ASCII  /                SEIZE   ERROR/<CR><LF>
9918  066662 020040 020040 020040
9919  066670 020040 020040 042523
9920  066676 055111 020105 020040
9921  066704 051105 047522 006522
9922  066712    012
9923  066713    124 051505 020124          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD/
9924  066720 020043 042440 051122
9925  066726 050040 020103 050040
9926  066734 051117 020124 020043
9927  066742 050040 051117 020124
9928  066750 020043 051040 043505
9929  066756 040440 051104 043440
9930  066764 047517 020104 020040
9931  066772 041040 042101    000
9932  066777    124 051505 020124  DH5:    .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR GOOD    BAD
9933  067004 020043 042440 051122
9934  067012 050040 020103 050040
9935  067020 051117 020124 020043
9936  067026 051040 043505 040440
9937  067034 051104 043440 047517
9938  067042 020104 020040 041040
9939  067050 042101    000
9940  067053    040 020040 020040  DH7:    .ASCII  /                RELSNG  ERROR/<CR><LF>
9941  067060 020040 020040 020040
9942  067066 020040 020040 051040
9943  067074 046105 047123 020107
9944  067102 042440 051122 051117
9945  067110 005015
9946  067112 042524 052123 021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR GOOD    BAD
9947  067120 020040 051105 020122
9948  067126 041520 020040 047520
9949  067134 052122 021440 020040
9950  067142 047520 052122 021440
9951  067150 020040 042522 020107
9952  067156 042101 020122 047507
9953  067164 042117 020040 020040
9954  067172 040502 000104
9955  067176 042524 052123 021440  DH11:   .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS
9956  067204 020040 051105 020122
```

```
 9957  067212  041520  020040  047520
 9958  067220  052122  021440  020040
 9959  067226  042522  020107  042101
 9960  067234  020122  047503  052116
 9961  067242  047105  051524     000
 9962  067247     040  020040  020040  DH13:   .ASCII                    SEIZE   ERROR/<CR><LF>
 9963  067254  020040  020040  020040
 9964  067262  020040  020040  051440
 9965  067270  044505  042532  020040
 9966  067276  042440  051122  051117
 9967  067304  005015
 9968  067306  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #  REG ADR CONTENTS/
 9969  067314  020040  051105  020122
 9970  067322  041520  020040  047520
 9971  067330  052122  021440  020040
 9972  067336  047520  052122  021440
 9973  067344  020040  042522  020107
 9974  067352  042101  020122  047503
 9975  067360  052116  047105  051524
 9976  067366     000
 9977  067367     040  020040  020040  DH22:   .ASCII  /                  RELSNG  SEIZE/<CR><LF>
 9978  067374  020040  020040  020040
 9979  067402  020040  020040  051040
 9980  067410  046105  047123  020107
 9981  067416  051440  044505  042532
 9982  067424  005015
 9983  067426  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
 9984  067434  020040  051105  020122
 9985  067442  041520  020040  047520
 9986  067450  052122  021440  020040
 9987  067456  047520  052122  021440
 9988  067464     000
 9989  067465     040  020040  020040  DH23:   .ASCII  /                  SEIZE/<CR><LF>
 9990  067472  020040  020040  020040
 9991  067500  020040  020040  051440
 9992  067506  044505  042532  005015
 9993  067514  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  REG ADR CONTENTS/
 9994  067522  020040  051105  020122
 9995  067530  041520  020040  047520
 9996  067536  052122  021440  020040
 9997  067544  042522  020107  042101
 9998  067552  020122  047503  052116
 9999  067560  047105  051524     000
10000  067565     040  020040  020040  DH26:   .ASCII  /                  RELSNG/<CR><LF>
10001  067572  020040  020040  020040
10002  067600  020040  020040  051040
10003  067606  046105  047123  006507
10004  067614     012
10005  067615     124  051505  020124          .ASCIZ  /TEST #  ERR PC  PORT #
10006  067622  020043  042440  051122
10007  067630  050040  020103  050040
10008  067636  051117  020124  000043
10009  067644  020040  020040  020040  DH31:   .ASCII                    RELSNG  RQSTNG  CR  LF
10010  067652  020040  020040  020040
10011  067660  020040  020040  042522
.0012  067666  051514  043516  020040
```

```
10013  067674  050522  052123  043516
10014  067702  005015
10015  067704  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10016  067712  020040  051105  020122
10017  067720  041520  020040  047520
10018  067726  052122  021440  020040
10019  067734  047520  052122  021440
10020  067742     000
10021  067743     124  051505  020124  DH36:   .ASCIZ  /TEST #  ERR PC  PORT #/
10022  067750  020043  042440  051122
10023  067756  050040  020103  050040
10024  067764  051117  020124  000043
10025  067772  042524  052123  021440  DH42:   .ASCIZ  /TEST #  ERR PC/
10026  070000  020040  051105  020122
10027  070006  041520     000
10028  070011     040  020040  020040  DH44:   .ASCII                  RELSNG  ERROR/<CR><LF>
10029  070016  020040  020040  020040
10030  070024  020040  020040  051040
10031  070032  046105  047123  020107
10032  070040  042440  051122  051117
10033  070046  005015
10034  070050  042524  052123  021440          .ASCIZ  /TEST #  ERR PC  PORT #  PORT #/
10035  070056  020040  051105  020122
10036  070064  041520  020040  047520
10037  070072  052122  021440  020040
10038  070100  047520  052122  021440
10039  070106     000
10040  070107     040  020040  020040  DH46:   .ASCII                  PORT A  PORT B/<CR><LF>
10041  070114  020040  020040  020040
10042  070122  020040  020040  050040
10043  070130  051117  020124  020101
10044  070136  050040  051117  020124
10045  070144  006502     012
10046  070147     124  051505  020124          .ASCIZ  /TEST #  ERR PC  RPDS1   RPDS1/
10047  070154  020043  042440  051122
10048  070162  050040  020103  051040
10049  070170  042120  030523  020040
10050  070176  051040  042120  030523
10051  070204     000
10052  070205     124  051505  020124  DH55:   .ASCIZ  /TEST #  ERR PC  PORT #  TIMEOUT VALUE (IN MS)/
10053  070212  020043  042440  051122
10054  070220  050040  020103  050040
10055  070226  051117  020124  020043
10056  070234  052040  046511  047505
10057  070242  052125  053040  046101
10058  070250  042525  024040  047111
10059  070256  046440  024523     000
10060  070263     044  050122  042101  DH56:   .ASCIZ  /$RPADR/
10061  070270  000122
10062
10063                                  .EVEN
10064
10065  070272  001242  001116  001234  DT1:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$BDDAT,0
10066  070300  001122  001126  000000
10067  070306  001242  001116  001122  DT3:    .WORD   TSTNUM,$ERRPC,$BDADR,$GDDAT,$BDDAT,0
10068  070314  001124  001126  000000
```

```
10069  070322  001242  001116  001234  DT5:    .WORD   TSTNUM,$ERRPC,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10070  070330  001122  001124  001126
10071  070336  000000
10072  070340  001242  001116  001240  DT6:    .WORD   TSTNUM,$ERRPC,OPPRT,$BDADR,$BDDAT,0
10073  070346  001122  001126  000000
10074  070354  001242  001116  001236  DT7:    .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$GDDAT,$BDDAT,0
10075  070362  001234  001122  001124
10076  070370  001126  000000
10077  070374  001242  001116  001236  DT13:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,$BDADR,$BDDAT,0
10078  070402  001234  001122  001126
10079  070410  000000
10080  070412  001242  001116  001236  DT22:   .WORD   TSTNUM,$ERRPC,SEIZPT,PTNBR,0
10081  070420  001234  000000
10082  070424  001242  001116  001236  DT23:   .WORD   TSTNUM,$ERRPC,SEIZPT,$BDADR,$BDDAT,0
10083  070432  001122  001126  000000
10084  070440  001242  001116  001236  DT31:   .WORD   TSTNUM,$ERRPC,SEIZPT,OPPRT,0
10085  070446  001240  000000
10086  070452  001242  001116  001236  DT36:   .WORD   TSTNUM,$ERRPC,SEIZPT,0
10087  070460  000000
10088  070462  001242  001116  001234  DT37:   .WORD   TSTNUM,$ERRPC,PTNBR,0
10089  070470  000000
10090  070472  001242  001116  000000  DT42:   .WORD   TSTNUM,$ERRPC,0
10091  070500  001242  001116  001170  DT46:   .WORD   TSTNUM,$ERRPC,$TMP2,$TMP3,0
10092  070506  001172  000000
10093  070512  001242  001116  001240  DT54:   .WORD   TSTNUM,$ERRPC,OPPRT,SEIZPT,0
10094  070520  001236  000000
10095  070524  001242  001116  001236  DT55:   .WORD   TSTNUM,$ERRPC,SEIZPT,TIME,0
10096  070532  001252  000000          DT56:   .WORD   $RPADR,0
10097  070536  001300  000000
10098
10099  070542     000     000     000  DF1:    .BYTE   0,0,0,0,0
10100  070545     000     000
10101  070547     000     000     000  DF5.    .BYTE   0,0,0,0,0,0
10102  070552     000     000     000
10103  070555     000     000     000  DF7:    .BYTE   0,0,0,0,0,0,0
10104  070560     000     000     000
10105  070563     000
10106  070564     000     000     000  DF31:   .BYTE   0,0,0,0
10107  070567     000
10108  070570     000     000     000  DF36:   .BYTE   0,0,0
10109  070573     000     000          DF42:   .BYTE   0,0
10110  070575     000     000     000  DF55:   .BYTE   0,0,0,1
10111  070600     001
10112  070601     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  070602  003070          TSTADR: .WORD   TST1            ;STARTING ADDRESS OF TEST 1
10126  070604  004470                  .WORD   TST2            ;STARTING ADDRESS OF TEST 2
10127  070606  006070                  .WORD   TST3            ;STARTING ADDRESS OF TEST 3
10128  070610  007470                  .WORD   TST4            ;STARTING ADDRESS OF TEST 4
10129  070612  010626                  .WORD   TST5            ;STARTING ADDRESS OF TEST 5
10130  070614  011764                  .WORD   TST6            ;STARTING ADDRESS OF TEST 6
10131  070616  012754                  .WORD   TST7            ;STARTING ADDRESS OF TEST 7
10132  070620  013744                  .WORD   TST10           ;STARTING ADDRESS OF TEST 10
10133  070622  014366                  .WORD   TST11           ;STARTING ADDRESS OF TEST 11
10134  070624  015010                  .WORD   TST12           ;STARTING ADDRESS OF TEST 12
10135  070626  016366                  .WORD   TST13           ;STARTING ADDRESS OF TEST 13
10136  070630  017744                  .WORD   TST14           ;STARTING ADDRESS OF TEST 14
10137  070632  021246                  .WORD   TST15           ;STARTING ADDRESS OF TEST 15
10138  070634  022550                  .WORD   TST16           ;STARTING ADDRESS OF TEST 16
10139  070636  024010                  .WORD   TST17           ;STARTING ADDRESS OF TEST 17
10140  070640  024676                  .WORD   TST20           ;STARTING ADDRESS OF TEST 20
10141  070642  025564                  .WORD   TST21           ;STARTING ADDRESS OF TEST 21
10142  070644  026744                  .WORD   TST22           ;STARTING ADDRESS OF TEST 22
10143  070646  030124                  .WORD   TST23           ;STARTING ADDRESS OF TEST 23
10144  070650  031134                  .WORD   TST24           ;STARTING ADDRESS OF TEST 24
10145  070652  032144                  .WORD   TST25           ;STARTING ADDRESS OF TEST 25
10146  070654  033352                  .WORD   TST26           ;STARTING ADDRESS OF TEST 26
10147  070656  034560                  .WORD   TST27           ;STARTING ADDRESS OF TEST 27
10148  070660  036554                  .WORD   TST30           ;STARTING ADDRESS OF TEST 30
10149  070662  037250                  .WORD   TST31           ;STARTING ADDRESS OF TEST 31
10150  070664  040510                  .WORD   TST32           ;STARTING ADDRESS OF TEST 32
10151  070666  041750                  .WORD   TST33           ;STARTING ADDRESS OF TEST 33
10152  070670  043040                  .WORD   TST34           ;STARTING ADDRESS OF TEST 34
10153  070672  044130                  .WORD   TST35           ;STARTING ADDRESS OF TEST 35
10154  070674  044744                  .WORD   TST36           ;STARTING ADDRESS OF TEST 36
10155  070676  045560                  .WORD   TST37           ;STARTING ADDRESS OF TEST 37
10156  070700  046542                  .WORD   TST40           ;STARTING ADDRESS OF TEST 40
10157  070702  047524                  .WORD   TST41           ;STARTING ADDRESS OF TEST 41
10158  070704  051132                  .WORD   TST42           ;STARTING ADDRESS OF TEST 42
10159  070706  052540                  .WORD   TST43           ;STARTING ADDRESS OF TEST 43
10160  070710  053422                  .WORD   TST44           ;STARTING ADDRESS OF TEST 44
10161  070712  054310                  .WORD   TST45           ;STARTING ADDRESS OF TEST 45
10162  070714  055256                  .WORD   TST46           ;STARTING ADDRESS OF TEST 46
10163
10164                                  ;ATTENTION BIT TABLE
10165
10166  070716     001          ATABIT: .BYTE   1               ;ATTENTION BIT FOR DRIVE 0
10167  070717     002                  .BYTE   2               ;ATTENTION BIT FOR DRIVE 1
10168  070720     004                  .BYTE   4               ;ATTENTION BIT FOR DRIVE 2
10169  070721     010                  .BYTE   10              ;ATTENTION BIT FOR DRIVE 3
10170  070722     020                  .BYTE   20              ;ATTENTION BIT FOR DRIVE 4
10171  070723     040                  .BYTE   40              ;ATTENTION BIT FOR DRIVE 5
10172  070724     100                  .BYTE   100             ;ATTENTION BIT FOR DRIVE 6
10173  070725     200                  .BYTE   200             ;ATTENTION BIT FOR DRIVE 7
10174
10175  070726  000046          MAXTN:  .WORD   $TN-1           ;MAXIMUM TEST NUMBER
10176
10177          000001                  .END
```

```
ABS     = 000200  1384#
ACL     = 000040  1419#  1428#
ACU     = 100000  1373#
ADDRIS    062364  2069    9465#
ADRERR    062137  1996    9437#
AOE     = 001000  1290#
ASR1      001232  1560#  2021*  5905  6179  7618  7832  8025  8026  8139  8140
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   5070   5097  5101  5112  5114  5118  5141  5145  5291  5295  5440  5444
                  5595   5630   5634  5784  5819  5823  5960  5995  5999  6066  6101  6105  6196
                  6200   6345   6347  6351  6369  6373  6396  6402  6419  6423  6547  6549  6553
                  6571   6575   6598  6604  6621  6625  6771  6775  6940  6944  7069  7073  7206
                  7210   7316   7318  7322  7339  7343  7366  7370  7464  7466  7470  7487  7491
                  7514   7518   7637  7672  7676  7851  7886  7890  8056  8060  8083  8089  8170
                  8174   8197   8203  8344  8348  8509  8513
BABIT     070716  2021   10166#
BIT0    = 000001  1310#
BIT1    = 000002  1311#
BIT2    = 000004  1312#
BIT3    = 000010  1313#
BIT4    = 000020  1314#
BIT5    = 000040  1315#
BIT6    = 000100  1316#
BIT7    = 000200  1317#
BIT8    = 000400  1208#
BIT9    = 001000  1209#
BADNO     062324  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    8695
BIT1    = 000002  1181#
BIT10   = 002000  1162#  8739
BIT11   = 004000  1161#  8702
BIT12   = 010000  1160#
BIT13   = 020000  1159#  8746
BIT14   = 040000  1158#  8677
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    8611#
CKCLK1  056514  8611    8620#
CKCLK2  056556  8621    8628#
CKCLK3  056566  8619    8627    8630#
CKEPR   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*   5091*   5104*   5108*   5121*   5256*   5269*   5405*   5418*   5563*   5571*   5580
                5752*   5760*   5769    5933*   5946*   6039*   6052*   6308*   6321*   6327*   6335*   6341*   6354*
                6413*   6426*   6510*   6523*   6529*   6537*   6543*   6556*   6615*   6628*   6696*   6709*   6711
                6726*   6734*   6738*   6746*   6865*   6878*   6880    6895*   6903*   6907*   6915*   7312*   7325*
                7333*   7346*   7460*   7473*   7481*   7494*   7714*   7722*   7731*   7739*   7748*   7756*   7928*
                7936*   7945*   7953*   7962*   7970*   8293*   8301*   8303    8313*   8321*   8323    8458*   8466*
                8468    8478*   8486*   8488
CKSWR  = 104407  8676    8734    8754    9407#
CLOCK   056576  8615    8624    8635#
CLR    = 000040  1228#   2146    2167    3688    3892    4090    4291    4531    6714    6883
CR     = 000015  1097#   1550#   8877    8887    9420    9423    9432    9437    9441    9445    9449    9459    9463
                9465    9917    9940    9962    9977    9989    10000   10009   10028   10040
CRLF   = 000200  1098#   8848    8887
CSF    = 000002  1360#   1378#
CSL    = 000010  1362#   1380#
DCK    = 100000  1296#
DCL    = 000100  1420#   1429#
DCU    = 000001  1426#
DCISP  = 177570  1104#   1526    1966
DEI    = 000040  1267#
DFF20  = 000002  1263#
DF1     070542  1610    1617    1624    1666    1736    1743    1829    1836    1864    1878    1906    10099#
DF31    070564  1729    1759    1780    1857    1871    1913    10106#
DF36    070570  1673    1715    1722    1751    1773    1794    1801    1808    1815    1822    10108#
DF42    070573  1843    1850    10109#
DF5     070547  1638    1645    1659    1680    1687    1787    1885    1892    1899    10101#
DF55    070575  1920    10110#
DF56    070601  1927    10112#
DF7     070555  1631    1652    1694    1701    1708    1766    10103#
DH1     066534  1608    1615    1827    1862    9903#
DH11    067176  1664    9955#
DH13    067247  1643    1678    1685    1883    1897    9962#
DH22    067367  1727    1757    1911    9977#
DH23    067465  1734    1741    1834    1876    1904    9989#
DH2E    067565  10000#
DH3     066605  1622    9910#
DH31    067644  1778    10009#
```

```
DH36     067743          1671    1713    1720    1749    1771    1792    1799    1806    1813    1820    10021#
DH4      066654          1629    9917#
DH42     067772          1841    1848    10025#
DH44     070011          1855    10028#
DH46     070107          1869    10040#
DH5      066777          1636    1657    1785    1890    9932#
DH55     070205          1918    10052#
DH56     070263          1925    10060#
DH6      067053          1650    1692    1699    1706    1764    9940#
DIGB   = 000004          1264#
DISPLA   001142          1526#   1966#   1974#   8716#   8738#
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    5070    5137    5241    5287    5390    5436
                         5543    5595    5626    5732    5784    5815    5937    5960    5991    6043    6066    6097    6192
                         6767    6936    7065    7202    7362    7510    7637    7668    7851    7882    8052    8166    8340
                         8505
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    5070    5137    5241    5287    5390    5436
                         5543    5595    5626    5732    5784    5815    5937    5960    5991    6043    6066    6097    6192
                         6767    6936    7065    7202    7306    7362    7454    7510    7637    7668    7851    7882    8052
                         8166    8340    8505
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      070272          1609    1616    1665    1863    1905    10065#
DT13     070374          1644    1679    1686    1884    1898    10077#
DT22     070412          1728    1758    10080#
DT23     070424          1735    1742    1828    1835    1877    10082#
DT3      070306          1623    10067#
DT31     070440          1779    1856    10084#
DT36     070452          1750    1772    1793    1800    1814    10086#
DT37     070462          1672    1714    1721    1807    1821    10088#
DT42     070472          1842    1849    10090#
DT46     070500          1870    10091#
DT5      070322          1637    1658    1786    1891    10069#
DT54     070512          1912    10093#
DT55     070524          1919    10095#
DT56     070536          1926    10097#
DT6      070340          10072#
DT7      070354          1630    1651    1693    1730    1707    1765    10074#
```

```
DVA    = 004000            1258#
ECH    = 000100            1287#
ECI    = 004000            1401#
EMTVEC= 000030             1192#    1951*    1952*
EM1      062476            1607     9485#
EM10     063103            1656     9540#
EM11     063133            1663     9546#
EM12     063217            1670     9556#
EM13     063267            1677     9564#
EM14     063354            1684     9574#
EM15     063417            1691     9581#
EM16     063476            1698     9590#
EM17     063551            1705     9599#
EM2      062517            1614     9489#
EM20     063630            1712     9608#
EM21     063710            1719     9617#
EM22     063763            1726     9626#
EM23     064050            1733     9636#
EM24     064116            1740     9644#
EM25     064175            1748     9653#
EM26     064240            1756     9660#
EM27     064325            1763     9670#
EM3      062541            1621     9494#
EM30     064403            1770     9679#
EM31     064500            1777     9691#
EM32     064555            1784     9700#
EM33     064626            1791     9708#
EM34     064730            1798     9720#
EM35     065033            1805     9733#
EM36     065112            1812     9742#
EM37     065164            1819     9750#
EM4      062623            1628     9504#
EM40     065232            1826     9758#
EM41     065307            1833     9767#
EM42     065351            1840     9774#
EM43     065437            1847     9785#
EM44     065514            1854     9794#
EM45     065571            1861     9803#
EM46     065650            1868     9812#
EM47     065717            1875     9820#
EM5      062654            1635     9510#
EM50     066005            1882     9831#
EM51     066070            1889     9841#
EM52     066153            1896     9851#
EM53     066251            1903     9863#
EM54     066332            1910     9873#
EM55     066425            1917     9884#
EM56     066472            1924     9892#
EM6      062722            1642     9518#
EM7      063022            1649     9530#
EMTERR   062107            1991     9432#
ERR    = 040000            1276#    6312     6514
ERRVEC= 000004             1185#    1963     1964*    1975*    8611*    8612*    8621*    8630*    8682     8683*    8685*    8688*
EXEC     002602            2027     2034#    2115     2313     2550     2788     2967     3143     3288     3431     3511     3602     3806
                           4011     4212     4411     4589     4711     4841     5022     5200     5349     5502     5691     5877     6156
                           6253     6455     6664     6833     6999     7136     7276     7424     7582     7796     7999     8113     8243
```

```
                    3408    8561
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    = ****** L   1474    8574    8581    9399    9400    9401    9402    9403    9405    9407    9408    9409    9410
                    9411    9412
GO     = 000001     1252#
GRV    = 000010     1265#
GTSWR  = 104406     1986    9405#
HCE    = 000200     1288#
HCI    = 002000     1400#
HCRC   = 000400     1289#
I      = 000011     1095#   8846    8887
IAE    = 002000     1291#
IE     = 000100     1206#
ILF    = 000001     1281#
ILR    = 000002     1282#
INTVEC = 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#   5019    5023#
                    5197    5201#   5346    5350#   5499    5503#   5688    5692#   5874    5878#   6153    6157#   6250
                    6254#   6452    6456#   6661    6665#   6830    6834#   6996    7000#   7133    7137#   7273    7277#
                    7421    7425#   7579    7583#   7793    7797#   7996    8000#   8110    8114#   8240    8244#   8405
                    8409#   8559
LF     = 000012     1096#   1551#   8881    8887    9420    9423    9432    9437    9441    9445    9449    9456    9459
                    9463    9465    9473    9917    9940    9962    9977    9989    10000   10009   10028   10040
LST    = 002000     1272#
MAPTN   070726      2052    10175#
MCLK   = 000002     1301#
MCPE   = 020000     1211#   5575    5764    7726    7743    7760    7940    7957    7974
MHS    = 001000     1368#   1386#
MINX   = 000004     1302#
MCH    = 020000     1334#
MCL    = 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    5070    5137    5241    5287    5390    5436
                    5543    5595    5626    5732    5784    5815    5935    5960    5991    6043    6066    6097    6192
                    6767    6936    7065    7202    7362    7510    7637    7668    7851    7382    8052    9166    8340
                    8505
MPE    = 000400     1231#
MRC    = 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#
NMS   = 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    5057#   5124#   5168
                        5173    5318    5323    5467    5472    5657    5662    5846    5851    6022    6027    6128    6133
                        6223    6228    6359#   6396    6402    6407#   6561#   6598    6604    6609#   6798    6803    6967
                        6972    7096    7101    7233    7238    7393    7398    7541    7546    7699    7704    7913    7918
                        8039#   8083    8089    8094#   8153#   8197    8203    8208#   8371    8376    8536    8541
NOCLOC  062235          2028    9449#
NOSEIZ  001246          1566#
NTRH11  062444          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#   5048#
                        5228#   5377#   5530#   5719#   5949#   6055#   6278#   6480#   6689#   6858#   7025#   7162#   7606#
                        7820#   8269#   8434#   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    5070    5137
                        5241    5287    5390    5436    5543    5595    5626    5732    5784    5815    5937    5960    5991
                        6043    6066    6097    6192    6767    6936    7065    7202    7362    7510    7637    7668    7851
                        7882    8052    8166    8340    8505
PIP   = 020000          1275#
PIRQ  = 177772          1102#
PIRQVE= 000240          1196#
PLC   = 020000          1372#   1389#
PDATA   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    2925    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    4745    4795
```

```
                    4797    4805    4851    4867    4868    4869    4892    4893    4903    4925    4926    4947    4948
                    4957    4976    4978    4986    5032    5045    5046    5063    5064    5076    5077    5089    5090
                    5138    5157    5159    5167    5210    5226    5227    5228    5250    5251    5288    5307    5309
                    5317    5359    5372    5373    5387    5388    5427    5428    5437    5456    5458    5466    5512
                    5528    5529    5530    5556    5557    5598    5599    5609    5617    5618    5627    5646    5648
                    5656    5701    5714    5715    5729    5730    5777    5778    5790    5791    5816    5835    5837
                    5845    5887    5900    5906    5907    5949    5963    5964    5974    5982    5983    5992    6011
                    6013    6021    6037    6038    6054    6059    6060    6072    6073    6098    6117    6119    6127
                    6166    6193    6212    6214    6222    6263    6275    6276    6302    6303    6366    6385    6387
                    6395    6431    6465    6480    6492    6493    6568    6587    6589    6597    6613    6614    6674
                    6686    6687    6715    6716    6736    6737    6758    6759    6768    6787    6789    6797    6843
                    6858    6893    6894    6937    6956    6958    6966    7009    7022    7023    7066    7085    7087
                    7095    7146    7162    7186    7187    7203    7222    7224    7232    7286    7294    7295    7296
                    7353    7354    7363    7382    7384    7392    7434    7475    7476    7511    7530    7532    7540
                    7592    7603    7604    7613    7614    7624    7625    7630    7631    7643    7644    7669    7688
                    7690    7698    7806    7820    7824    7825    7833    7834    7854    7855    7865    7873    7874
                    7883    7902    7904    7912    8009    8017    8053    8072    8074    8082    8123    8134    8143
                    8144    8145    8157    8158    8167    8186    8188    8196    8253    8269    8270    8271    8312
                    8331    8332    8341    8360    8362    8370    8418    8431    8432    8442    8443    8506    8525
                    8527    8535
PORTA:  062161       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    5036    5048    5049
                    5050    5073    5074    5084    5106    5107    5128    5129    5142    5153    5154    5172    5214
                    5223    5224    5238    5239    5278    5279    5292    5303    5304    5322    5363    5375    5376
                    5377    5399    5400    5441    5452    5453    5471    5516    5525    5526    5540    5541    5588
                    5589    5601    5602    5631    5642    5643    5661    5705    5717    5718    5719    5745    5746
                    5787    5788    5798    5806    5807    5820    5831    5832    5850    5891    5914    5915    5921
                    5922    5948    5953    5954    5966    5967    5996    6007    6008    6026    6055    6069    6070
                    6080    6088    6089    6102    6113    6114    6132    6170    6197    6208    6209    6227    6267
                    6278    6290    6291    6370    6381    6382    6401    6411    6412    6469    6477    6478    6504
                    6505    6572    6583    6584    6603    6633    6678    6689    6724    6725    6772    6783    6784
                    6802    6847    6855    6856    6884    6885    6905    6906    6927    6928    6941    6952    6953
                    6971    7013    7025    7049    7050    7070    7081    7082    7100    7150    7159    7160    7207
                    7218    7219    7237    7290    7327    7328    7367    7378    7379    7397    7438    7442    7443
                    7444    7501    7502    7515    7526    7527    7545    7596    7606    7610    7611    7619    7620
                    7640    7641    7651    7659    7660    7673    7684    7685    7703    7810    7817    7818    7827
                    7828    7838    7839    7844    7845    7857    7858    7887    7898    7899    7917    8013    8020
                    8029    8030    8031    8043    8044    8057    8068    8069    8088    8127    8131    8171    8182
                    8183    8202    8257    8266    8267    8277    8278    8345    8356    8357    8375    8422    8434
                    8435    8436    8477    8496    8497    8510    8521    8522    8540
PORTBI  062207       2010    9445#
PORTC   001230       1559#   2017#   2018#   2019#   5899    6174
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#
PSW     = 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*  5050*  5064*  5074*  5077*  5084*  5090*
                          5107*  5129*  5153*  5157*  5167*  5172*  5227*  5239*  5251*  5279*  5303*  5307*  5317*
                          5322*  5376*  5388*  5400*  5428*  5452*  5456*  5466*  5471*  5529*  5541*  5557*  5589*
                          5599*  5602*  5609*  5618*  5642*  5646*  5656*  5661*  5718*  5730*  5746*  5778*  5788*
                          5791*  5798*  5807*  5831*  5835*  5845*  5850*  5907*  5915*  5922*  5954*  5964*  5967*
                          5974*  5983*  6007*  6011*  6021*  6026*  6038*  6060*  6070*  6073*  6080*  6089*  6113*
                          6117*  6127*  6132*  6208*  6212*  6222*  6227*  6291*  6303*  6381*  6385*  6395*  6401*
                          6412*  6493*  6505*  6583*  6587*  6597*  6603*  6614*  6716*  6725*  6737*  6759*  6783*
                          6787*  6797*  6802*  6885*  6894*  6906*  6928*  6952*  6956*  6966*  6971*  7050*  7081*
                          7085*  7095*  7100*  7187*  7218*  7222*  7232*  7237*  7295*  7328*  7354*  7378*  7382*
                          7392*  7397*  7443*  7476*  7502*  7526*  7530*  7540*  7545*  7611*  7614*  7620*  7625*
                          7631*  7641*  7644*  7651*  7660*  7684*  7688*  7698*  7703*  7825*  7828*  7834*  7839*
                          7845*  7855*  7858*  7865*  7874*  7898*  7902*  7912*  7917*  8030*  8044*  8068*  8072*
                          8082*  8088*  8144*  8158*  8182*  8186*  8196*  8202*  8271*  8278*  8312*  8332*  8356*
                          8360*  8370*  8375*  8436*  8443*  8477*  8497*  8521*  8525*  8535*  8540*  10065  10069
                         10074  10077  10080  10088
PWRVEC= 000024            1191#
RAW     = 000020          1381#
RDCHR = 104410            9216    9408#
RDLIN = 104411            9289    9409#
RDOCT = 104412            1992    2048    2074    9410#
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*  5069*  5134*
                          5162*  5284*  5312*  5433*  5461*  5594*  5623*  5651*  5783*  5812*  5840*  5959*  5988*
                          6016*  6065*  6094*  6122*  6189*  6217*  6362*  6390*  6564*  6592*  6764*  6792*  6933*
                          6961*  7062*  7090*  7199*  7227*  7359*  7387*  7507*  7535*  7636*  7665*  7693*  7950*
                          7879*  7907*  8049*  8077*  8163*  8191*  8337*  8365*  8502*  8530*
RELOK = 000001            1931#  2441    2678    2931    3110    3205    3350    3482    3562    3578#  3765   3969   4151
                          4352   4511    4679    4801    4982    5163    5313    5462    5652    5841   6017   6123   6218
                          6391   6593    6793    6962    7091    7228    7388    7536    7694    7908   8078   8192   9366
                          8531
RESREG= 104414            9412#
RESVEC= 000010            1186#
```

RMR = 000004
RPAS = 000016
RPBA = 000004
RPCA = 000034
RPCC = 000036
RPCS1 = 000000
RPCS2 = 000010

1283#
1454# 5905* 6181* 7618* 7832* 8023 8033 8137 8147
1449#
1461# 2373 2497 2610 2734
1462# 2372 2502 2609 2739
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* 5034* 5035* 5038* 5039* 5065* 5130* 5163* 5164* 5212* 5213* 5216* 5217*
5225 5257 5258 5280* 5313* 5314* 5361* 5362* 5365* 5366* 5374 5406 5407
5429* 5462* 5463* 5514* 5515* 5518* 5519* 5558* 5572 5573 5578* 5590* 5619*
5652* 5653* 5703* 5704* 5707* 5708* 5747* 5761 5762 5767* 5779* 5808* 5841*
5842* 5889* 5890* 5893* 5894* 5955* 5984* 6017* 6018* 6061* 6090* 6123* 6124*
6168* 6169* 6172* 6173* 6218* 6219* 6265* 6266* 6269* 6270* 6391* 6392* 6433*
6434* 6467* 6468* 6471* 6472* 6593* 6594* 6635* 6636* 6676* 6677* 6680* 6681*
6695* 6697 6698 6717* 6752* 6760* 6793* 6794* 6845* 6846* 6849* 5850* 6964*
6866 6867 6886* 6921* 6929* 6962* 6963* 7011* 7012* 7015* 7016* 7091* 7092*
7148* 7149* 7152* 7153* 7228* 7229* 7288* 7289* 7292* 7293* 7301* 7355* 7363*
7389* 7436* 7437* 7440* 7441* 7449* 7503* 7536* 7537* 7594* 7595* 7598* 7599*
7632* 7661* 7694* 7695* 7723 7724 7729* 7740 7741 7746* 7757 7758 7763*
7800* 7809* 7812* 7813* 7846* 7875* 7908* 7909* 7937 7938 7943* 7954 7955
7960* 7971 7972 7977* 8011* 8012* 8015* 8016* 8045* 8078* 8079* 8125* 8126*
8129* 8130* 8159* 8192* 8193* 8255* 8256* 8259* 8260* 8283* 8333* 8366* 8367*
8420* 8421* 8424* 8425* 8448* 8498* 8531* 8532*
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* 5032* 5036* 5045* 5049* 5063* 5073* 5076*
5089* 5106* 5128* 5138* 5142* 5154* 5159* 5210* 5214* 5223* 5226* 5238* 5250*
5278* 5298* 5292* 5304* 5309* 5359* 5363* 5372* 5375* 5387* 5399* 5427* 5437*
5441* 5453* 5458* 5512* 5516* 5525* 5528* 5540* 5556* 5588* 5598* 5601* 5617*
5627* 5631* 5643* 5648* 5701* 5705* 5714* 5717* 5729* 5745* 5777* 5787* 5790*
5806* 5816* 5820* 5832* 5837* 5887* 5891* 5899* 5906* 5914* 5921* 5953* 5963*
5966* 5982* 5992* 5996* 6008* 6013* 6037* 6059* 6069* 6072* 6088* 6098* 6102*
6114* 6119* 6166* 6170* 6174* 6193* 6197* 6209* 6214* 6263* 6267* 6275* 6290*
6302* 6366* 6370* 6382* 6387* 6411* 6431* 6465* 6469* 6477* 6492* 6504* 6568*
6572* 6584* 6589* 6613* 6633* 6674* 5678* 6686* 6714* 6715* 6724* 6736* 6758*
6768* 6772* 6784* 6789* 6843* 6847* 6855* 6883* 6884* 6893* 6905* 6927* 6937*
6941* 6953* 6958* 7009* 7013* 7022* 7049* 7066* 7070* 7082* 7087* 7146* 7150*
7159* 7186* 7203* 7207* 7219* 7224* 7286* 7290* 7294* 7327* 7353* 7363* 7367*
7379* 7384* 7434* 7438* 7442* 7475* 7501* 7511* 7515* 7527* 7532* 7592* 7596*
7603* 7610* 7613* 7619* 7624* 7630* 7640* 7643* 7659* 7669* 7673* 7685* 7690*
7806* 7810* 7817* 7824* 7827* 7833* 7838* 7844* 7854* 7857* 7873* 7883* 7887*
7899* 7904* 8009* 8013* 8017* 8020* 8029* 8043* 8053* 8057* 8069* 8074* 8123*
8127* 8134* 8134* 8143* 8157* 8167* 8171* 8183* 8188* 8253* 8257* 8266* 8270*

```
                    8277*  8331*  8341*  8345*  8357*  8362*  8418*  8422*  8431*  8435*  8442*  8496*  8506*
                    8510*  8522*  8527*
RPDA  = 000006      1450#  2378   2472   2615   2709
RPDB  = 000022      1456#
RPDS1 = 000012      1452#  2128   2149   2213   2214   2228   2229   2245   2246   2260   2261   2337*  234:
                    2343   2352   2385   2413   2417   2421   2574*  2578   2580   2589   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   5033*  5037*  5047*  5055*  5071   5075   5078   5092   5093   5109
                    5110   5135   5139   5143   5211*  5215*  5229   5231   5240   5285   5289   5293   5360*
                    5364*  5378   5380   5389   5434   5438   5442   5513*  5517*  5527*  5531   5533   5542
                    5564   5565   5596   5600   5603   5624   5628   5632   5702*  5706*  5716*  5720   5722
                    5731   5753   5754   5785   5789   5792   5813   5817   5821   5888*  5892*  5912   5916
                    5923   5934   5935   5961   5965   5968   5989   5993   5997   6040   6041   6067   6071
                    6074   6095   6099   6103   6167*  6171*  6190   6194   6198   6264*  6268*  6277*  6296
                    6309   6310   6342   6343   6363   6367   6371   6414   6415   6432*  6466*  6470*  6479*
                    6498   6511   6512   6544   6545   6565   6569   6573   6616   6617   6634*  6675*  6679*
                    6688*  6727   6728   6765   6769   6773   6844*  6848*  6857*  6896   6897   6934   6938
                    6942   7010*  7014*  7024*  7048   7051   7063   7067   7071   7147*  7151*  7161*  7185
                    7188   7200   7204   7208   7287*  7291*  7306   7313   7314   7334   7335   7360   7364
                    7368   7435*  7439*  7454   7461   7462   7482   7483   7508   7512   7516   7593*  7597*
                    7605*  7638   7642   7645   7666   7670   7674   7807*  7811*  7819*  7852   7856   7859
                    7880   7884   7888   8010*  8014*  8021   8032*  8050   8054   8058   8124*  8128*  8135
                    8146*  8164   8168   8172   8254*  8258*  8268*  8276*  8294   8295   8314   8315   8338
                    8342   8346   8419*  8423*  8433*  8441*  8459   8460   8479   8480   8503   8507   8511
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*  6283*  6328   6329   6485*  6530   6531   6694*  6739   6740
                    6863*  6908   6909   7607*  7617*  7621*  7715   7716   7821*  7831*  7835*  7929   7930
                    8018*  8019*  8132*  8133*
RPER2 = 000040      1463#  2375   2487   2612   2724   7608*  7616*  7622*  7732   7733   7822*  7830*  7836*
                    7946   7947
RPER3 = 000042      1464#  2370   2512   2607   2749   7609*  7615*  7623*  7749   7750   7823*  7829*  7837*
                    7963   7964
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      9411#
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#  5046#  5224#  5373#  5526#  5715#  5900#  5948#  6054#  6276#  6478#
                  6687#  6856#  7023#  7160#  7296#  7444#  7604#  7818#  8031#  8145#  8267#  8432#  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   5028   5206   5355   5508   5697   5883
                  6162   6259   6461   6670   6839   7005   7142   7282   7430   7588   7802   8005   8119
                  8249   8414
START     002064  1491   1937#
START1    002072  1495   1939#
START2    002100  1938   1940#
STKLMT= 177774  1101#
SWR       001140  1525#  1945   1965#  1967   1973#  1984   2285   8677   8695   8702   8739   8746   8751
                  9068   9107   9154#
SWREG     000176  1476#  1973   1984   9068   9107   9130
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#
TRAP    = 040000  1335#
TBITVE= 000014  1187#
TDF     = 000040  1364#  1382#
TESTNO    062304  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          5025    5026    5028#
TEST23  030176          5203    5204    5206#
TEST24  031206          5352    5353    5355#
TEST25  032216          5505    5506    5508#
TEST26  033424          5694    5695    5697#
TEST27  034632          5880    5881    5883#
TEST3   006142          2553    2554    2556#
TEST30  036626          6159    6160    6162#
TEST31  037322          6256    6257    6259#
TEST32  040562          6458    6459    6461#
TEST33  042022          6667    6668    6670#
TEST34  043112          6836    6837    6839#
TEST35  044202          7002    7003    7005#
TEST36  045016          7139    7140    7142#
TEST37  045632          7279    7280    7282#
TEST4   007542          2791    2792    2794#
TEST40  046614          7427    7428    7430#
TEST41  047576          7585    7586    7588#
TEST42  051204          7799    7800    7802#
TEST43  052612          8002    8003    8005#
TEST44  053474          8116    8117    8119#
TEST45  054362          8246    8247    8249#
TEST46  055330          8411    8412    8414#
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#   6288#   6490#
                        7034#   7042#   7056    7171#   7179#   7193    9635#   10095
TIMEA   001256          1570#   2022#   2321#   2392#   2394
*TIMEAM 001262          1572#   2323#   2396#   7111
*TIMEAP 001260          1571#   2023#   2322#   2395#   7109    8288
*TIMEB  001264          1573#   2024#   2558#   2629#   2631
*TIMEBM 001270          1575#   2560#   2633#   7248
*TIMEBP 001266          1574#   2025#   2559#   2632#   7246    8453
TIMES   001272          1576#   7056#   7109    7111    7193#   7246    7248
*TITLE  062010          1979    9420#
TKVEC = 000060          1194#   9053#   9054#
*OLER   056630          2393    2630    8645#
TPVEC = 000064          1195#
*PAPVE= 000034          1193#   1953#   1954#
*RE   = 040000          1212#   5578    5767    7729    7746    7763    7943    7960    7977
TRK1  = 004000          1351#
TRK10 = 040000          1354#
TRK2  = 010000          1352#
TRK20 = 100000          1355#
TRK4  = 020000          1353#
*RTVEC= 000014          1188#
*STADR  070602          2058    10125#
TSTERR  062352          9463#
*STNUM  001242          1564#   9735#   10065   10067   10069   10072   10074   10077   10080   10082   10084   10086   10089
                        10090   10091   10093   10095
*ST1    003070          2051    2111#   10125
*ST1AA  003064          2096#   8597
```

```
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          5018#  10142
TST23   030124          5196#  10143
TST24   031134          5345#  10144
TST25   032144          5498#  10145
TST26   033352          5687#  10146
TST27   034560          5873#  10147
TST3    006070          2546#  10127
TST30   036554          6152#  10148
TST31   037250          6249#  10149
TST32   040510          6451#  10150
TST33   041750          6660#  10151
TST34   043040          6829#  10152
TST35   044130          6995#  10153
TST36   044744          7132#  10154
TST37   045560          7272#  10155
TST4    007470          2784#  10128
TST40   046542          7420#  10156
TST41   047524          7578#  10157
TST41B  050600          7612    7626    7713#
TST42   051132          7709    7792#  10158
TST42B  052206          7826    7840    7927#
TST43   052540          7923    7995#  10159
TST44   053422          8109#  10160
TST45   054310          8239#  10161
TST46   055256          8404#  10162
TST5    010626          2963#  10129
TST6    011764          3139#  10130
TST7    012754          3284#  10131
TUF     = 000100        1365#   1383#
TYPOS   = 104405        8578    8585    8805    9403#
TYPE    = 104401        1979    1991    1996    2004    2010    2016    2028    2036    2047    2054    2069    2072    2073
                        8572    8579    8586    8741    8749    8769    8786    8788    8791    8793    8808    8813    8851
                        8947    9022    9074    9128    9129    9132    9145    9156    9175    9223    9229    9234    9238
                        9243    9244    9246    9249    9253    9318    9320    9399#
TYPOC   = 104402        2071    8777    8801    9131    9400#
TYPON   = 104404        9402#
TYPOS   = 104403        2007    2013    9401#
UNS     = 040000        1295#
UPE     = 020000        1236#
US1     = 000001        1223#
US2     = 000002        1224#
US4     = 000004        1225#
USRF    = 000010        1417#
VUF     = 000002        1416#
VUSC    = 010000        1371#
```

```
.$$     = 000100  1268#  2419  2423  2444  2450  2656  2660  2681  2687  2871  2909  2913  2934
                  2940  3050  3084  3088  3092  3179  3183  3187  3324  3328  3332  3456  3460
                  3464  3536  3540  3544  3632  3673  3716  3739  3743  3747  3836  3877  3920
                  3943  3947  3951  4125  4129  4133  4326  4330  4334  4489  4493  4630  4653
                  4657  4661  4752  4775  4779  4783  4889  4956  4960  4964  5070  5137  5141
                  5145  5241  5287  5291  5295  5390  5436  5440  5444  5543  5595  5626  5630
                  5634  5732  5784  5815  5819  5823  5937  5960  5991  5995  5999  6043  6066
                  6097  6101  6105  6192  6196  6200  6369  6373  6571  6575  6767  6771  6775
                  6936  6940  6944  7065  7069  7073  7202  7206  7210  7362  7366  7370  7510
                  7514  7518  7637  7668  7672  7676  7851  7882  7886  7890  8052  8056  8060
                  8166  8170  8174  8340  8344  8348  8505  8509  8513
.$$SET  = 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  5070  5137  5163  5165  5168  5173
                  5241  5287  5313  5315  5318  5323  5390  5436  5462  5464  5467  5472  5543
                  5595  5626  5652  5654  5657  5662  5732  5784  5815  5841  5843  5846  5851
                  5960  5991  6017  6019  6022  6027  6066  6097  6123  6125  6128  6133  6192
                  6218  6223  6228  6391  6393  6396  6397  6402  6403  6593  6595  6598  6599
                  6604  6605  6767  6793  6795  6798  6803  6936  6962  6964  6967  6972  7065
                  7091  7096  7101  7202  7228  7233  7238  7362  7388  7390  7393  7398  7510
                  7536  7538  7541  7546  7637  7668  7694  7696  7699  7704  7851  7882  7908
                  7910  7913  7918  8052  8078  8080  8083  8084  8089  8090  8166  8192  8194
                  8197  8198  8203  8204  8340  8366  8368  8371  8376  8505  8531  8533  8536
                  8541
.MAC    = 000002  1427#
.MCALL  = 001254  1569#  2329*  2387  2566*  2624  2802*  2890  2981*  3069  3155*  3267  3300*  3412
                  6289*  6298  6491*  6500  7035*  7036  7043*  7053  7172*  7173  7180*  7190  8288*
                  8306  8453*  8471  8636  8638*  8640*
.MCE    = 040000  1237#
.MCF    = 000040  1286#
.MCL    = 000001  1359#  1377#
.MLE    = 004000  1292#
.MRL    = 004000  1273#
.MSL    = 000400  1367#  1385#
.MSL    = 000004  1361#  1379#
$ALTJB  = 001134  1522#  1988*  9121  9269
$BDACR  = 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*  5071*  5072*  5093*  5094*  5110*  5111*  5135*  5136*  5230*  5231*  5258*  5259*
```

```
                5285*  5286*  5379*  5380*  5407*  5408*  5434*  5435*  5532*  5533*  5565*  5566*  5573*
                5574*  5596*  5597*  5624*  5625*  5721*  5722*  5754*  5755*  5762*  5763*  5785*  5786*
                5813*  5814*  5935*  5936*  5961*  5962*  5989*  5990*  6041*  6042*  6067*  6068*  6095*
                6096*  6190*  6191*  6310*  6311*  6329*  6330*  6343*  6344*  6363*  6364*  6415*  6416*
                6512*  6513*  6531*  6532*  6545*  6546*  6565*  6566*  6617*  6618*  6698*  6699*  6728*
                6729*  6740*  6741*  6765*  6766*  6867*  6868*  6897*  6898*  6909*  6910*  6934*  6935*
                7063*  7064*  7200*  7201*  7314*  7315*  7335*  7336*  7360*  7361*  7462*  7463*  7483*
                7484*  7508*  7509*  7638*  7639*  7666*  7667*  7716*  7717*  7724*  7725*  7733*  7734*
                7741*  7742*  7750*  7751*  7758*  7759*  7852*  7853*  7880*  7881*  7930*  7931*  7938*
                7939*  7947*  7948*  7955*  7956*  7964*  7965*  7972*  7973*  8023*  8024*  8050*  8051*
                8137*  8138*  8164*  8165*  8295*  8296*  8315*  8316*  8338*  8339*  8460*  8461*  8480*
                8481*  8503*  8504*  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*  2579*  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*  5078*  5083*  5085
                5092*  5096   5100   5109*  5113   5117   5152*  5158*  5166*  5171*  5229*  5233   5240*
                5244   5257*  5261   5265   5302*  5308*  5316*  5321*  5378*  5382   5389*  5393   5406*
                5410   5414   5451*  5457*  5465*  5470*  5531*  5535   5542*  5546   5564*  5568   5572*
                5575   5603*  5608*  5610   5641*  5647*  5655*  5660*  5720*  5724   5731*  5735   5753*
                5757   5761*  5764   5792*  5797*  5799   5830*  5836*  5844*  5849*  5934*  5938   5942
                5968*  5973*  5975   6006*  6012*  6020*  6025*  6040*  6044   6048   6074*  6079*  6081
                6112*  6118*  6126*  6131*  6207*  6213*  6221*  6226*  6309*  6313   6317   6328*  6332
                6342*  6346   6350   6380*  6386*  6394*  6400*  6414*  6418   6422   6511*  6515   6519
                6530*  6534   6544*  6548   6552   6582*  6588*  6596*  6602*  6616*  6620   6624   6697*
                6701   6705   6727*  6731   6739*  6743   6782*  6788*  6796*  6801*  6866*  6870   6874
                6896*  6900   6908*  6912   6951*  6957*  6965*  6970*  7080*  7086*  7094*  7099*  7217*
                7223*  7231*  7236*  7313*  7317   7321   7334*  7338   7342   7377*  7383*  7391*  7396*
                7461*  7465   7469   7482*  7486   7490   7525*  7531*  7539*  7544*  7645*  7650*  7652
                7683*  7689*  7697*  7702*  7715*  7719   7723*  7726   7732*  7736   7740*  7743   7749*
                7753   7757*  7760   7859*  7864*  7866   7897*  7903*  7911*  7916*  7929*  7933   7937*
                7940   7946*  7950   7954*  7957   7963*  7967   7971*  7974   8033*  8034   8067*  8073*
                8081*  8087*  8147*  8148   8181*  8187*  8195*  8201*  8294*  8298   8314*  8319   8355*
                8361*  8369*  8374*  8459*  8463   8479*  8483   8520*  8526*  8534*  8539*  10065  10067
                10069  10072  10074  10077  10082
$BELL   001202  1545*  8741   8761   9074   9263
$CHGRC  057644  8853*  8863*  8870   8879*  8884*
$CKSWP  060574  9107*  9407
$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          9128    9264#
$CNTLU   061431          9145    9238    9263#
$CRLF    001207          1547#   2016    2036    2072    8586    8749    8761    8769    8788    8793    8813    8852    8887
                         9156    9243    9263    9323
$DBLK    060312          8988    9022    9030#
$DOAGN   056434          8568    8589    8595#
$DTBL    060302          8991    9026#
$ENDAD   056424          1484    8591#   8756
$ENDCT   056270          1955    8570#
$ENULL   056440          8598#
$EOP     056224          8209    8558#
$EOPCT   056262          1955#   8567#   8571
$ERFLG   001103          1508#   8667    8691    8693    8699#   8720    8736#   8761
$ERMAX   001115          1514#   1958#   2044#   8693    8715#   8720
$ERROR   057122          1951    8733#
$ERRPC   001116          1515#   8743#   8744#   8745    8761    8775    10065   10067   10069   10072   10074   10077   10080
                         10082   10084   10086   10088   10090   10091   10093   10095
$ERRTB   001304          1603#   8783
$ERRTY   057254          8748    8758#
$ERTTL   001112          1512#   8583    8587#   8742#   8761
$ESCAP   001200          1544#   1957#   8714#
$FILLC   001156          1533#   8856    8887
$FILLS   001155          1532#   8887
$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#   3345    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#   4204#   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    5070#
                         5085    5095#   5098    5102#   5112#   5115    5119#   5137#   5168    5173    5232#   5233    5241#
                         5242    5246    5260#   5263    5267#   5287#   5318    5323    5381#   5382    5390#   5391    5395
                         5409#   5412    5416#   5436#   5467    5472    5534#   5535    5543#   5544    5548    5567#   5568
                         5595#   5610    5626#   5657    5662    5723#   5724    5732#   5733    5737    5756#   5757    5784#
                         5799    5815#   5846    5851    5937#   5940    5944#   5960#   5975    5991#   6022    6027    6043#
                         6046    6050#   6066#   6081    6097#   6128    6133    6192#   6223    6228    6312#   6315    6319#
                         6331#   6332    6345#   6348    6352#   6365#   6397    6403    6417#   6420    6424#   6514#   6517
                         6521#   6533#   6534    6547#   6550    6554#   6567#   6599    6605    6619#   6622    6626#   6700#
                         6703    6707#   6730#   6731    6742#   6743    6767#   6798    6803    6869#   6872    6876#   6899#
                         6900    6911#   6912    6936#   6967    6972    7065#   7096    7101    7202#   7233    7238    7316#
                         7319    7323#   7337#   7340    7344#   7362#   7393    7398    7464#   7467    7471#   7485#   7488
                         7492#   7510#   7541    7546    7637#   7652    7668#   7699    7704    7718#   7719    7735#   7736
                         7752#   7753    7851#   7866    7882#   7913    7918    7932#   7933    7949#   7950    7966#   7967
                         8025#   8036    8052#   8084    8090    8139#   8150    8166#   8198    8204    8297#   9298    8317#
```

```
                      8318    8340*   8371    8376    8462*   8463    8482*   8483    8505*   8536    8541    10067   10069
                     10074
$GET42  056414        8588*
$GTSWR  060664        9129*   9405
$HD     = 000000      1074
$HIOCT  061624        9311*   9322*
$ICNT   001104        1509*   2060*   8706*   8707    8709*   8719
$INTAG  001135        1523*   9126*   9157    9269
$ITEMB  001114        1513*   8745*   8761    8772
$LF     001210        1548*   8761    8887    9253    9263    9323
$LKCSB  001214        1553*   8617*
$LKCSR  001212        1552*   8613    8618*
$LKS    001220        1555*   8622    8626*
$LLVEC  001222        1556*   8623
$LPADR  001106        1510*   1959*   2045*   2118*   2316*   2553*   2791*   2970*   3146*   3291*   3434*   3514*   3605*
                      3809*   4014*   4215*   4414*   4592*   4714*   4844*   5025*   5203*   5352*   5505*   5694*   5880*
                      6159*   6256*   6458*   6667*   6836*   7002*   7139*   7279*   7427*   7585*   7799*   8002*   8116*
                      8246*   8411*   8697*   8712*   8717    8719
$LPERR  001110        1511*   1960*   2046*   2119*   2317*   2554*   2792*   2971*   3147*   3292*   3435*   3515*   3606*
                      3810*   4015*   4216*   4415*   4593*   4715*   4845*   5026*   5204*   5353*   5506*   5695*   5881*
                      6160*   6257*   6459*   6668*   6837*   7003*   7140*   7280*   7428*   7586*   7800*   8003*   8028*
                      8117*   8142*   8247*   8412*   8697    8713*   8719
$LPVEC  001216        1554*   8614
$MAIL = ****** U      1977    1984    8712    8751    8840
$MNEW   061454        9132    9267*
$MSWR   061443        9129    9265*
$MXCNT  057120        8710    8719*
$NULL   001154        1531*   8858    8887
$NWAIT= 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    5180*   5182    5329*
                      5331    5479*   5481    5668*   5670    5858*   5860    6140*   6142    6235*   6237    6437*   6439
                      6640*   6642    6809*   6811    6979*   6981    7116*   7118    7255*   7257    7404*   7406    7553*
                      7555    7767*   7769    7982*   7984    8096*   8098    8218*   8220    8383*   8385
$OCNT   060072        8919*   8948*   8961*
$OMODE  060074        8914*   8918*   8923    8926*   8937*   8963*
$OVER   057104        8678    8698    8708    8716*
$PASS   001100        1506*   2043*   8564*   8565*   8576    8598    8704    8720
$QUES   001206        1546*   8761    8887    9175    9246    9263    9320    9323
$RDCHR  061076        9188*   9408
$RDDEC= ****** U      9411
$RDLIN  061166        9211*   9409
$RDOCT  061466        9284*   9410
$RDSZ = 000007        9204*
$REGAD  001160        1535*
$REGO   001162        1537*
$RESRE  061664        9356*   9412
$RPADR  001300        1586*   2037    2070    2077*   2080    2096    10097
$RPVEC  001302        1587*
$RTNAD  056436        8597*
$R2A  = ****** U      9413
$SAVRE  061626        9340*   9411
$SCOPE  056670        1949    8675*
$SETUP= 000127        1931*   1948    1949    1951    1953    1955    1956    1957    1959    1981    9562    8676    9734
                      8754    8756    9066    9071    9072    9102    9269
$SETUP = 177777       1931*
```

```
$SVLAD  057056          8686    8711#
$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    5027    5205    5354    5507    5696    5882
                        6161    6258    6460    6669    6838    7004    7141    7281    7429    7587    7801    8004    8118
                        8248    8413    8555    8563    8590    8596    8598    8668    8669    8670    8671    8672    8677
                        8689    8691    8692    8693    8700    8701    8702    8713    8716    8719    8720#   8726    8727
                        8728    8729    8730    8739    8746    8751    8755    8761
$SWRMK= 000000          8672
$TIMES  001176          1543#   1956*   2120*   2318*   2555*   2793*   2972*   3148*   3293*   3436*   3516*   3607*   3811*
                        4016*   4217*   4416*   4594*   4716*   4846*   5027*   5205*   5354*   5507*   5696*   5882*   6161*
                        6258*   6460*   6669*   6838*   7004*   7141*   7281*   7429*   7587*   7801*   8004*   8118*   8248*
                        8413*   8563*   8700*   8707    8710*   8719
$TKB    001146          1528#   9034    9055    9064    9083    9111    9138
$TKCNT  060322          9035#   9050*   9072    9089*   9195    9197*
$TKINT  060332          1980    2041    5050#   9124
$TKQEN= 060331          9039#   9097    9200
$TKQIN  060324          9036#   9051*   9052    9095*   9096*   9097    9099*
$TKQOU  060326          9037#   9052*   9198    9199*   9200    9202*
$TKQSR  060330          9038#   9051    9099    9202
$TKS    001144          1527#   9034    9056*   9079*   9081    9087*   9109    9125*   9135    9159*
$TKSRV  060402          9053    9064#
$TKPC   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*   3905*   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    5075*   5080    5083    5096*   5097*   5098    5113*   5114*   5115    5140*   5141*   5146
                        5148    5155    5244*   5245*   5246    5261*   5262*   5263    5290*   5291*   5296    5298    5305
                        5393*   5394*   5395    5410*   5411*   5412    5439*   5440*   5445    5447    5454    5546*   5547*
                        5548    5600*   5605    5608    5629*   5630*   5635    5637    5644    5735*   5736*   5737    5789*
                        5794    5797    5818*   5819*   5824    5826    5833    5938*   5939*   5940    5965*   5970    5973
                        5994*   5995*   6000    6002    6009    6044*   6045*   6046    6071*   6076    6079    6100*   6101*
                        6106    6108    6115    6179*   6180*   6181    6195*   6196*   6201    6203    6210    6313*   6314*
                        6315    6346*   6347*   6348    6368*   6369*   6374    6376    6383    6418*   6419*   6420    6515*
                        6516*   6517    6548*   6549*   6550    6570*   6571*   6576    6578    6585    6620*   6621*   6622
                        6701*   6702*   6703    6770*   6771*   6776    6778    6785    6870*   6871*   6872    6939*   6940*
                        6945    6947    6954    7068*   7069*   7074    7076    7083    7205*   7206*   7211    7213    7220
                        7317*   7318*   7319    7338*   7339*   7340    7365*   7366*   7371    7373    7380    7465*   7466*
                        7467    7486*   7487*   7488    7513*   7514*   7519    7521    7528    7642*   7647    7650    7671*
                        7672*   7677    7679    7686    7856*   7861    7864    7885*   7886*   7891    7893    7900    8034*
                        8035*   8036    8055*   8056*   8061    8063    8070    8148*   8149*   8150    8169*   8170*   8175
                        8177    8184    8343*   8344*   8349    8351    8358    8509*   8509*   8514    8516    8523
```

$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 5144# 5145# 5146 5160 5242# 5243# 5245 5294#
5295# 5296 5310 5391# 5392# 5394 5443# 5444# 5445 5459 5544# 5545# 5547
5633# 5634# 5635 5649 5733# 5734# 5736 5822# 5823# 5824 5838 5998# 5999#
6000 6014 6104# 6105# 6106 6120 6199# 6200# 6201 6215 6372# 6373# 6374
6388 6574# 6575# 6576 6590 6774# 6775# 6776 6790 6943# 6944# 6945 6959
7072# 7073# 7074 7088 7209# 7210# 7211 7225 7369# 7370# 7371 7385 7517#
7518# 7519 7533 7675# 7676# 7677 7691 7889# 7890# 7891 7905 8026# 8027#
8035 8059# 8060# 8061 8075 8140# 8141# 8149 8173# 8174# 8175 8189 8347#
8348# 8349 8363 8512# 8513# 8514 8528

$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 5139# 5140 5152 5166 5168 5289# 5290 5302 5316 5318 5438#
5439 5451 5465 5467 5628# 5629 5641 5655 5657 5817# 5818 5830 5844
5846 5993# 5994 6006 6020 6022 6099# 6100 6112 6126 6128 6194# 6195
6207 6221 6223 6367# 6368 6380 6394 6396# 6397 6569# 6570 6582 6596
6598# 6599 6769# 6770 6782 6796 6798 6938# 6939 6951 6965 6967 7067#
7068 7080 7094 7096 7204# 7205 7217 7231 7233 7364# 7365 7377 7391
7393 7512# 7513 7525 7539 7541 7670# 7671 7683 7697 7699 7884# 7885
7897 7911 7913 8054# 8055 8067 8081 8083# 8084 8168# 8169 8181 8195
8197# 8198 8342# 8343 8355 8369 8371 8507# 8508 8520 8534 8536 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 4689 4781# 4782 4796 4809 4811 4962# 4963 4977
4990 4992 5143# 5144 5158 5171 5173 5293# 5294 5308 5321 5323 5442#
5443 5457 5470 5472 5632# 5633 5647 5660 5662 5821# 5822 5836 5849
5851 5997# 5998 6012 6025 6027 6103# 6104 6118 6131 6133 6198# 6199
6213 6226 6228 6371# 6372 6386 6400 6402# 6403 6573# 6574 6588 6602
6604# 6605 6773# 6774 6788 6801 6803 6942# 6943 6957 6970 6972 7071#
7072 7086 7099 7101 7208# 7209 7223 7236 7238 7368# 7369 7383 7396
7398 7516# 7517 7531 7544 7546 7674# 7675 7689 7702 7704 7888# 7889
7903 7916 7918 8058# 8059 8073 8087 8089# 8090 8172# 8173 8187 8201
8203# 8204 8346# 8347 8361 8374 8376 8511# 8512 8526 8539 8541 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# 4194
4249# 4250# 4251 4269# 4270# 4271 4306# 4307# 4308 4383# 4384# 4395 4448#
4449# 4450 4468# 4469# 4470 4547# 4548# 4549 4564# 4565# 4566 4919# 4920#

```
                    4921    4936*   4937*   4938    5100*   5101*   5102    5117*   5118*   5119    5265*   5266*   5267
                    5414*   5415*   5416    5942*   5943*   5944    6048*   6049*   6050    6317*   6318*   6319    6350*
                    6351*   6352    6422*   6423*   6424    6519*   6520*   6521    6552*   6553*   6554    6624*   6625*
                    6626    6705*   6706*   6707    6874*   6875*   6876    7321*   7322*   7323    7342*   7343*   7344
                    7469*   7470*   7471    7490*   7491*   7492
$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    5018    5027*   5028    5180    5196    5205*   5206
                    5329    5345    5354*   5355    5479    5498    5507*   5508    5668    5687    5696*   5697    5858
                    5873    5882*   5883    6140    6152    6161*   6162    6235    6249    6258*   6259    6437    6451
                    6460*   6461    6640    6660    6669*   6670    6809    6829    6838*   6839    6979    6995    7004*
                    7005    7116    7132    7141*   7142    7255    7272    7281*   7282    7404    7420    7429*   743C
                    7553    7578    7587*   7588    7589    7709    7767    7792    7801*   7802    7803    7923    7982
                    7995    8004*   8005    8096    8109    8118*   8119    8218    8239    8248*   8249    8383    9404
                    8413*   8414    10175
$TPB    001152      1530*   8876*   8887
$TPFLG  001157      1534*   8834    8887
$TPS    001150      1529*   8874    8887
$TRAP   061722      1953    9376*
$TRAP2  061744      9387*   9398
$TPP    = 000015    9391*   9400*   9401*   9402*   9403*   9404*   9405    9406*   9407    9408*   9409*   9410*   9411*
                    9412*   9413*
$TRPAD  061756      9381    9398*
$TSTNM  001102      1507*   2117*   2315*   2552*   2790*   2969*   3145*   3290*   3433*   3513*   3604*   3808*   4013*
                    4214*   4413*   4591*   4713*   4843*   5024*   5202*   5351*   5504*   5693*   5879*   6158*   6255*
                    6457*   6666*   6835*   7001*   7138*   7278*   7426*   7584*   7798*   8001*   8115*   8245*   8410*
                    8562*   8667    8711*   8716    8720    8735    8738    8761
$TTYIN  061422      9213    9214    9226    9244    9258    9262*
$TYPBN= ******      9404
$TYPDS  060076      8976*   9403
$TYPE   057430      8834*   9391    9399
$TYPEC  057600      8855    8862    8869    8874*   8875    9161
$TYPEX  057646      8880    8882    8885*
$TYPOC  057674      8917*   9400
$TYPON  057710      8916    8919*   9402
$TYPOS  057650      8912*   9401
$XTSTR  056702      8680*
$$GET4= 000000      8590*
$CFILL  060073      8913*   8917*   8927    8962*
$40CAT= ******      8677    8748
.       = 070730    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    5581    5770
                    5924    6712    6881    7112    7249    7307    7455    8304    8307    8324    8469    8472    8489
                    8560    8575*   8598    8599*   8719    8720    8761    8816*   8887    9030*   9034    9038*   9039
                    9040*   9262*   9263    9269*   9323
```

```
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  2842  2867  2954  3006
            3021   3046  3224  3241  3369  3386  3552  3669  3695  3712  3656  3873  3899  3916  4039
            4059   4096  4173  4240  4260  4297  4374  4429  4459  4538  4555  4910  4927  5091  5108
            5256   5405  5563  5752  5932  6039  6308  6327  6341  6413  6510  6529  6543  6615  6696
            6726   6738  6865  6895  6907  7312  7333  7460  7481  7714  7731  7748  7928  7945  7962
            8293   8313  8458  8478
CLRATA      1063#  4596  4718  4848  5029  5207  5356  5509  5639  5884  6163  6260  6426  6462  6630
            6671   6840  7006  7143  7283  7431  7589  7803  8006  8120  8250  8415
COMMEN      1197#
DD          4596#  4718
EE          4848#  5029
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
            3246   3253  3269  3337  3350  3356  3362  3381  3398  3414  3469  3482  3488  3494  3549
            3562   3568  3574  3626  3635  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  5082  5087  5103  5120  5150  5165  5170  5175
            5235   5248  5268  5300  5315  5320  5325  5384  5397  5417  5449  5464  5469  5474  5537
            5550   5570  5577  5607  5612  5639  5654  5659  5664  5726  5739  5759  5766  5796  5801
            5828   5843  5848  5853  5918  5926  5945  5972  5977  6004  6019  6024  6029  6051  6078
            6083   6110  6125  6130  6135  6205  6220  6225  6230  6300  6320  6334  6353  6378  6393
            6399   6405  6425  6502  6522  6536  6555  6580  6595  6601  6607  6627  6708  6733  6745
            6780   6795  6800  6805  6877  6902  6914  6949  6964  6969  6974  7055  7079  7093  7098
            7103   7113  7192  7215  7230  7235  7240  7250  7324  7345  7375  7390  7395  7400  7472
            7493   7523  7538  7543  7548  7649  7654  7681  7696  7701  7706  7721  7728  7738  7745
            7755   7762  7863  7868  7895  7910  7915  7920  7935  7942  7952  7959  7969  7976  8038
            8065   8080  8086  8092  8152  8179  8194  8200  8206  8300  8320  8353  8368  8373  8379
            8465   8485  8518  8533  8538  8543
ESCAPE      1197#
FF          5207#  5356
FF1         5509#  5698
GETPRI      1197#
GET_SWP     1197#  1981
GG          5884#
HEAD        1063#  2111  2303  2546  2784  2963  3139  3284  3427  3507  3598  3802  4007  4208  4407
            4585   4707  4837  5018  5196  5345  5498  5687  5873  6152  6249  6451  6660  6829  6995
            7132   7272  7420  7578  7792  7995  8109  8239  8404
HH          6163#
II          6260#  6462
JJ          6671#  6840
KK          7006#  7143
LL          7589#  7803
MM          7589#  7709  7923
NN          7283#  7431
OOO         4418#
```

```
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   5179#  5182   5328#  5331   5478#  5481   5667#  5670   5857#
             5860   6139#  6142   6234#  6237   6436#  6439   6639#  6642   6808#  6811   6979#  6981   7115#  7118
             7254#  7257   7403#  7406   7552#  7555   7766#  7769   7981#  7984   8095#  8098   8217#  8220   8382#
             8385
MULT         1197#
NEGTRA       1063#  2409   2646   2898   3077   3173   3318   3450   3530   3733   3937   4119   4320   4479   4647
             4769   4950   5131   5281   5430   5620   5809   5985   6091   6186   6359   6561   6761   6930   7059
             7196   7356   7504   7662   7876   8046   8160   8334   8499
NEWTST       1197#  2098   2291   2528   2766   2947   3127   3272   3418   3498   3579   3783   3988   4190   4393
             4573   4695   4818   4998   5180   5329   5479   5668   5858   6140   6235   6437   6640   6809   6979
             7116   7255   7404   7553   7767   7982   8096   8218   8383
NN           8250#  8415
OC           3609#  3813
PCP          1197#  9017   9312   9361
PUSH         1197#  8976   9286   9341
RELEAS       1063#  3167   3312   3727   3931   4113   4314   4641   4763   4879   4944   5060   5125   5275   5424
             5585   5614   5774   5803   5950   5979   6056   6085   6755   6924   7350   7498   7622   7656   7841
             7970   8039   8153   8328   8493
REPORT       1147#
RR           8006#  8120
SCOPE        1092#  2288   2526   2763   2945   3124   3270   3415   3496   3576   3781   3985   4138   4389   4570
             4693   4815   4996   5177   5327   5476   5666   5855   6137   6232   6435   6637   6807   6976   7114
             7251   7402   7550   7708   7922   8094   8208   8381   9546
SEIZE        1063#  2332   2569   3158   3303   3611   3815   4079   4280   4609   4731   4861   5042   5220   5369
             5522   5711   6272   6474   6683   6852   7019   7156   7600   7814   8263   9428
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   5049   5063   5073   5076   5089
             5106   5128   5226   5237   5250   5278   5375   5386   5399   5427   5528   5539   5556   5588   5595
             5601   5617   5717   5728   5745   5777   5787   5790   5806   5906   5914   5920   5953   5963   5966
             5982   6036   6059   6069   6072   6088   6290   6301   6411   6492   6503   6613   6715   6723   6736
             6758   6884   6892   6905   6927   7049   7186   7294   7327   7353   7442   7475   7501   7610   7613
             7619   7624   7630   7640   7643   7659   7824   7827   7833   7838   7844   7854   7857   7873   8029
             8043   8143   8157   8270   8277   8331   8435   8442   8496
SETATA       1063#  4019   4220   4419
SETPRI       1197#  9191
SETTRA       9391#  9400   9401   9402   9403   9405   9407   9408   9409   9410   9411   9412
SETUP        1197#  1941
SKIP         1197#
SLASH        1197#
SPACE        1197#
STABS        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   3298   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   4960   4871   4876   4943   4998   5017   5041   5052
```

```
                5057    5124    5180    5195    5219    5253    5272    5329    5344    5368    5402    5421    5479    5497    5521
                5553    5560    5584    5668    5686    5710    5742    5749    5773    5858    5872    5896    5902    5909    5929
                6033    6140    6151    6176    6183    6235    6248    6271    6280    6285    6293    6305    6324    6338    6358
                6408    6437    6450    6473    6482    6487    6495    6507    6526    6540    6560    6610    6640    6659    6682
                6691    6720    6749    6754    6809    6828    6851    6860    6889    6918    6923    6979    6994    7018    7027
                7031    7039    7045    7058    7106    7116    7131    7155    7164    7168    7176    7182    7195    7243    7255
                7271    7298    7303    7309    7330    7349    7404    7419    7446    7451    7457    7478    7497    7553    7577
                7767    7791    7982    7994    8096    8108    8211    8215    8218    8238    8262    8273    8280    8285    8290
                8309    8327    8383    8403    8427    8438    8445    8450    8455    8474    8492    8550    8601    8605    8656
                8660    8664    8722    8763    8819    8889    8966    9033    9102    9117    9180    9204    9272    9325    9370
                9414    9418    9479    9483   10117   10121
SWITCH          1063#   4885    5066    5591    5780    5956    6062    7633    7847
SWRSU           1197#   1961#
TESTAG          1063#   2121    2319    2556    2794    2973    3149    3294    3437    3517    3608    3812    4017    4218    4417
                4595    4717    4847    5028    5206    5355    5508    5697    5883    6162    6259    6461    6670    6839    7005
                7142    7282    7430    7588    7802    8005    8119    8249    8414
TIMER           1063#   2324    2561    2797    2976    3150    3295    6284    6486    7030    7038    7167    7175
TRMTRP          9391#
TYPBIN          1197#
TYPDEC          1197#   8576    8583    8803
TYPNAM          1197#
TYPNUM          1197#
TYPOCS          1197#   2005    2011
TYPOCT          1197#   8775    8800    9130
TYPTXT          1197#   8572    8579
USERFO          8548#   8559
$$CMRE          1498#   1537
$$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    5180    5329    5479    5668    5858    6140    6235    6437    6640    6809    6979
                7116    7255    7404    7553    7767    7982    8096    8218    8383
$$SET           9391#   9400    9401    9402    9403    9405    9407    9408    9409    9410    9411    9412
$$SKIP          1197#
.EQUAT          1063#   1087
.HEADE          1063#   1064
.SETUP          1063#   1931
.SWRHI          1063#   1075
.SWRLO          1063#   1085#
.$ACTI          1063#   1478
.$CATC          1063#   1468
.$CMTA          1063#   1498
.$EOP           1063#   8548
.$ERRO          1063#   8720
.$ERRT          1063#   8761
.$RDOC          1063#   9270
.$READ          1063#   9031
.$SAVE          1063#   9323
.$SCOP          1063#   8662
.$TRAP          1063#   9368
.$TYPD          1063#   8964
.$TYPE          1063#   8817
.$TYPO          1063#   8887

. ABS.  070730    900
```

ERRORS DETECTED: 0

RM03:CZRJEB,CZRJEB.SEQ/CRF/SOL/LI:ME/NL:MC:MD:CND=CZRJEB.P11
RUN-TIME: 36 37 3 SECONDS
RUN-TIME RATIO: 215/77=2.7
CORE USED: 31K (61 PAGES)

D01